



---

JOMAR: Joint Operations with Mobile Autonomous Robots

Edwin Olson  
UNIVERSITY OF MICHIGAN

---

12/21/2015  
Final Report

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory  
AF Office Of Scientific Research (AFOSR)/ IOA  
Arlington, Virginia 22203  
Air Force Materiel Command

<b>REPORT DOCUMENTATION PAGE</b>		<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>			
<b>1. REPORT DATE (DD-MM-YYYY)</b> 14-10-2015		<b>2. REPORT TYPE</b> Final	
		<b>3. DATES COVERED (From - To)</b> 20 Apr 2011 - 19 Jun 2015	
<b>4. TITLE AND SUBTITLE</b> JOMAR: Joint Operations with Mobile Autonomous Robots		<b>5a. CONTRACT NUMBER</b> FA23861114024	
		<b>5b. GRANT NUMBER</b> Grant AOARD-114024	
		<b>5c. PROGRAM ELEMENT NUMBER</b> 61102F	
<b>6. AUTHOR(S)</b> Dr. Edwin Olson		<b>5d. PROJECT NUMBER</b>	
		<b>5e. TASK NUMBER</b>	
		<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> University of Michigan 2260 Hayward Street Ann Arbor, MI 48109 United States		<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b> N/A	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> AOARD UNIT 45002 APO AP 96338-5002		<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b> AFRL/AFOSR/IOA(AOARD)	
		<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b> AOARD-114024	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Distribution A: Approved for public release. Distribution is unlimited			
<b>13. SUPPLEMENTARY NOTES</b>			
<b>14. ABSTRACT</b> Under this grant, we formulated and implemented a variety of novel algorithms that address core problems in multi-robot systems. These contributions roughly fall into two categories: state estimation and mapping, and robot perception and computer vision. 1. State Estimation and Mapping: * Methods for performing non-linear optimization with non-Gaussian error models. This provides a fundamental advantage over Gaussian methods which are unable to model real world sensor failure modes. This MaxMixture formulation is the stand-out success of this grant based on adoption and citation by the community. * A characterization of Global Positioning System (GPS) noise models in the MaxMixture framework, allowing significant improvements in GPS-aided navigation. * A data-association algorithm with applications to target tracking and computer vision applications, named the Incremental Posterior Joint Compatibility (IPJC) test, which computes optimal data associations in a small fraction of the time required by previous methods. 2. Robot Perception and Computer Vision * A method for learning visual features based on the needs of an application. Previous approaches rely on humans to design high-performance visual features; we show for the first time that such filters can be learned in-situ. * A new camera calibration system that achieves dramatically more accurate and consistent calibration results than previous methods. 3. Radio Communication and Mesh Networking * New methods for predicting the signal strength between two robots in a mesh network leveraging both previous robot radio communication attempts and non-radio sensor data such as LIDAR.			

**15. SUBJECT TERMS**

Autonomous Agents and Multi-Agent Systems, Multi-Agent Technology, Unmanned Vehicles

**16. SECURITY CLASSIFICATION OF:****a. REPORT****b. ABSTRACT****c. THIS PAGE**

U

U

U

**17. LIMITATION OF  
ABSTRACT**

SAR

**18. NUMBER  
OF PAGES**

72

**19a. NAME OF RESPONSIBLE PERSON**

Brian Lutz, Lt Col, USAF

**19b. TELEPHONE NUMBER** *(Include area code)*  
+81-42-511-2006**Standard Form 298 (Rev. 8/98)**

Prescribed by ANSI Std. Z39.18

# Joint Operations of Multiple Autonomous Robots (JOMAR)

**PI:** Edwin Olson (University of Michigan)

**Project Period:** 4/20/2011 to 6/20/2015

**Project Period Award:** \$750,000.00

**Project Objective:** Formulate and implement methods allowing robust operation of multi-robot teams, particularly A) state estimation and B) computer vision. Also investigating several other areas; see pubs list.

## Highlights

Under this grant, we formulated and implemented a variety of novel algorithms that address core problems in multi-robot systems. These contributions roughly fall into two categories: state estimation and mapping, and robot perception and computer vision.

### 1. State Estimation and Mapping:

- \* Methods for performing non-linear optimization with non-Gaussian error models. This provides a fundamental advantage over Gaussian methods which are unable to model real-world sensor failure modes. This “MaxMixture” formulation is the stand-out success of this grant based on adoption and citation by the community.

- \* A characterization of Global Positioning System (GPS) noise models in the MaxMixture framework, allowing significant improvements in GPS-aided navigation.

- \* A data-association algorithm with applications to target tracking and computer vision applications, named the Incremental Posterior Joint Compatibility (IPJC) test, which computes optimal data associations in a small fraction of the time required by previous methods.

### 2. Robot Perception and Computer Vision

- \* A method for learning visual features based on the needs of an application. Previous approaches rely on humans to design high-performance visual features; we show for the first time that such filters can be learned in-situ.

- \* A new camera calibration system that achieves dramatically more accurate and consistent calibration results than previous methods.

### 3. Radio Communication and Mesh Networking

\* New methods for predicting the signal strength between two robots in a mesh network leveraging both previous robot radio communication attempts and non-radio sensor data such as LIDAR.

## New Robotics TestBed

Largely developed using equipment budget from this grant, we developed a second-generation multi-robot test bed. This testbed is based on our first-place MAGIC 2010 robot design, but has a completely re-designed drive train and electronics suite. The new design doubles the ground speed to around 6mph and greatly increases the terrain handling capability as well. We have built six of these robots and have plans to build additional robots. We have begun talking to other researchers interested in using this platform in their own research.



## Evaluation Site

Over the course of this project, we spent a total of 10 days doing real-world evaluation of our system at the Muscatatuck Urban Training Center (MUTC) in Indiana. Muscatatuck offers a variety of venues, including a subterranean maze complex (top) and a shanty town (bottom) which our robots collaboratively mapped.



## Tier 1 Peer-Reviewed Publications

Publication	Citations
[1] Johannes Strom and Edwin Olson. Multi-sensor ATTenuation Estimation (MATTE): Signal-strength prediction for teams of robots. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2012.	4
[2] Edwin Olson and Yangming Li. IPJC: The Incremental Posterior Joint Compatibility Test for Fast Feature Cloud Matching. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2012.	7
[3] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. International Journal of Robotics Research, 826-840, July 2013.	65
[4] Andrew Richardson and Edwin Olson. Learning Convolutional Filters for Interest Point Detection. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), May 2013.	4
[5] Andrew Richardson, Johannes Strom and Edwin Olson. AprilCal: Assisted and repeatable camera calibration. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), November 2013.	6
[6] Ryan Morton and Edwin Olson. Robust Sensor Characterization via Max-Mixture Models: GPS Sensors. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), November 2013.	3

# Multi-sensor ATTenuation Estimation (MATTE): Signal-strength prediction for teams of robots

Johannes Strom and Edwin Olson

**Abstract**—Multi-robot teams are often constrained by communications; better signal-strength models enable more efficient coordination while still maintaining adequate communication. This work discusses several prediction algorithms applicable to this scenario. Whereas previous approaches typically focus on prediction in the presence of deployed base-stations, we consider the more general problem where all nodes in the network can be mobile. Our new algorithm, Multi-sensor ATTenuation Estimation (MATTE), addresses this problem by leveraging other forms of sensor data in combination with signal-strength measurements to infer the locations of attenuating materials in the robots’ environment. We also extend prior tomographic and correlation-based approaches to the multi-robot case, allowing a competitive evaluation. All methods are evaluated on a large corpus of real-world indoor and outdoor environments.

## I. INTRODUCTION

The inherently parallel nature of search-and-rescue missions creates an opportunity for teams of collaborating robots to assist in emergency response. However, robot teams that wish to collaborate must communicate to coordinate effectively. Current approaches to multi-robot coordination typically only incorporate a simple fixed-radius communication model as a planning constraint [1], [2], [3]. In environments with variable attenuation (e.g. urban environments), picking a single radius may not be appropriate since communication will be easier in open spaces, and harder in densely-built neighborhoods.

Attempts to incorporate more complicated models of signal propagation typically focus on the case where robots communicate with an existing fixed base station [4]. Unfortunately, many domains lack a usable communications infrastructure (e.g. disaster zones), forcing robots to deploy their own. Achieving good performance from fully-mobile networks is challenging because more complicated signal-propagation models must be incorporated into the path-planning process to ensure connectivity. Furthermore, existing models for fixed-transmitters do not extend directly to the case where all nodes are mobile. While prediction in the former case is analogous to regression in a two dimensional space, the latter requires making predictions for a four dimensional space, but without a corresponding increase in data density. The result is that achieving similar generalization performance from observed signal-strength measurements becomes more challenging. This paper explores how existing methods can be modified to better cope with this reduction

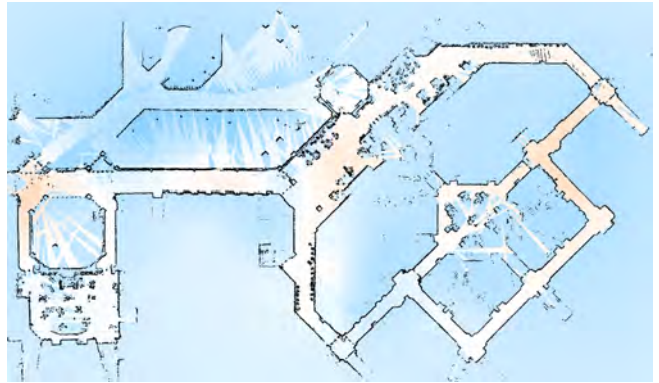


Fig. 1. Attenuation estimation computed from a single traversal of a  $160 \times 100\text{m}$  environment by 3 robots conducting an exploration mission. Blue indicates regions where signals are attenuated, white and orange regions indicate where signals pass more easily. Black denotes building structure. Our algorithm, MATTE, is designed for predicting signal strength in the challenging case where all transmitters and receivers are mobile.

in data density, and explores methods for using additional sensor data to inform a better prior over the locations of significant attenuators in the environment. Specifically, the main contributions of this paper are:

- Extension of tomographic and correlative signal prediction techniques to the case of multiple mobile robots without a base station.
- A new signal-strength prediction method, Multi-sensor ATTenuation Estimation (MATTE), which additionally leverages the robots’ LIDAR data to better predict the location of attenuating objects.
- Extensive evaluation on real world datasets covering over  $40,000\text{ m}^2$  of both indoor and outdoor environments

## II. RELATED WORK

In the robotics domain, planning with guaranteed communication is a well studied topic [5], [6]. However, communication between two agents cannot be ensured in general, so these methods are limited in their application to the real world. Others have studied collaborative planning under a fixed-radius communication assumption [1], but such methods result in unnecessarily conservative strategies because they fail to exploit long-range links when possible, reducing the effective speed of the robots. More realistic models can be obtained by first predicting the signal strength along a given link, which can then be used to predict packet success rates. There are two main approaches to signal prediction in the literature – the first is *correlative* in nature, that is

This work was supported by U.S. DoD Grant FA2386-11-1-4024.

The authors are with Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48109 USA {jstrom, ebolson}@umich.edu



broadcasts from nearby positions are assumed to have similar signal strengths, allowing prediction at nearby points using what is essentially a locally-weighted average. The second technique uses principles from *tomography*, where changes in signal-strength are used to infer the presence of attenuating objects, which can in turn be used to predict signal-strength at unknown locations.

Malmirchegini and Mostofi have explored correlative methods of predicting the link quality between a fixed base station and a mobile robot, taking into account the spatial correlations of signal strength measurements [7]. Fink and Kumar use a similar approach to allow a robot to automatically localize a base-station [8]. However, these approaches, also demonstrated in a simulated robotic domain in [4], assume signals are uniformly correlated in all directions without regard to the location of attenuating objects. Nonetheless, this approach has successfully been applied to prediction of signal strength in real-world environments, and forms the basis for one of the methods we benchmark in this paper. Our extension of their approach addresses the more general problem when all nodes are potentially mobile. This makes the problem significantly more challenging because of a substantial reduction in data density.

An alternative approach to predicting signal strength, using principles from the field of tomography, is to explicitly estimate the location and properties of attenuating objects affecting signal propagation. Knowing the location of attenuating objects enables future signal strength to be predicted, even when they are not spatially adjacent to previous measurements. The central challenge with this technique is that directly estimating the positions of attenuating objects from signal-strength measurements results in an ill-posed estimation problem. This is because for any given set of signal measurements, there are many possible world configurations which explain the data. Prior work by Wilson and Patwari has explored the capabilities of RF tomography to recover the motion of moving people inside a region bounded by a large number of regularly placed radios [9]. Their work estimated the derivative of the attenuation over a grid of pixels inside the perimeters of radios to extract the positions of moving people. By only estimating the derivative, their signal model was simplified, allowing a single calibration step to be used to later recover the target's position. Despite the large number of radios (28), their estimation problem was still ill-posed, requiring application of regularization techniques to make computing the attenuation-derivative possible [10]. However, their approach does not directly apply to the case where multiple robots traverse arbitrary paths, since in such cases no prior calibration is possible, and the number of links constraining the attenuation computed at each pixel can vary considerably. Our signal-strength prediction algorithm MATTE addresses these problems by employing a novel fusion of laser range-finder data with signal-strength measurements to better constrain the attenuations estimated at each pixel. Furthermore, we show that our approach scales to environments hundreds of meters in diameter.

### III. BACKGROUND

Standard macroscopic models of RF propagation describe the expected signal strength, which depends on the location of the receiver  $\vec{r}$  and transmitter  $\vec{t}$ , as having three main components [2]:

$$y_{dBm} = \underbrace{L_0 - a \log_{10}(\|\vec{r} - \vec{t}\|)}_{\text{path-loss}} - \underbrace{g(\vec{r}, \vec{t})}_{\text{shadowing}} - \underbrace{\epsilon}_{\text{multipath}} \quad (1)$$

In simplified, ideal environments, signal-strength can be determined purely by path-loss which has two degrees of freedom:  $L_0$  corresponds to the power of the transmitter and  $a$  is the path-loss exponent that determines how quickly the signal attenuates with distance. In real environments both shadowing and multipath can additionally affect the signal strength: shadowing corresponds to the attenuation a signal experiences as it passes through dense objects, and multipath corresponds to amplification or cancellation which occurs when waves travel multiple paths of different lengths from source to destination. In general, multipath is very difficult to predict because it results from complex reflection and diffraction interactions. Shadowing, on the other hand, is easier to predict, since the scale of its effects are larger and more spatially coherent. For the remainder of this paper, we will focus on models which predict path-loss and shadowing but ignore multipath.

Robots are able to measure the signal-strength from each other robot during the robot's mission. These noisy values form a vector  $\hat{y}$ , with corresponding vectors of positions  $R$  and  $T$  containing all pairs of receiver and transmitter positions, respectively. We treat predicting  $y$  as a linear regression problem. For the case of the simple path-loss model, estimating the two variables ( $L_0, a$ ) from the data is sufficient to predict signal measurements for arbitrary positions. This can be done using standard least-squares approaches: if  $x = [L_0 \ a]$  and the  $i^{\text{th}}$  row of  $A$  is  $[1 \ \log_{10}(\|\vec{r}_i - \vec{t}_i\|)]$  then the least-squared error estimate for  $x$  is  $\bar{x} = (A^T A)^{-1} A^T \hat{y}$ . Path-loss-only models are useful due to their simplicity and correspondence with theoretical propagation equations: the low degree of freedom reduces the chance of over-fitting. However, as this model ignores shadowing effects, its application in environments with varying attenuation is limited. In our evaluation, we will use this simple log-fit as a baseline.

Correlative and tomographic methods include the same path-loss model, but also explicitly incorporate shadowing, allowing for improved predictive performance. The way in which shadowing is captured varies between the two types of models. In the tomographic case, the shadowing function,  $g(\cdot)$  is computed by integrating the effect of all attenuators between transmitter and receiver. In practice, we model the individual attenuation of a grid of infinitely tall columns, represented by a regular 2D grid of pixels. If the signal passes through a series of  $p$  discretized pixels, the shadowing is computed as [9]:

$$g(\vec{r}, \vec{t}) = \sum_i^p w_i v_i \quad (2)$$

where  $v_i$  is the attenuation in the  $i^{\text{th}}$  pixel and  $w_i$  is the importance weight of that pixel for that signal's path. (e.g. in our implementation, the weights  $w_i$  correspond to the length of the line between transmitter and receiver which is contained in the pixel.) In other words, we compute attenuation per pixel by assuming that signal strength is reduced linearly according to the sum of the pixels along the path between receiver and transmitter. Given a set of signal strength measurements, we can attempt to find the attenuation values  $x_i$  of each pixel using a similar least-squares approach as described above. However, there are many possible attenuation assignments to the pixels which can adequately explain the signal strength measurements, resulting in an under-determined system of equations. In the next section we will discuss applying regularization techniques to encode a prior that prefers real-world environments, thereby over-constraining the system of equations.

In the correlative case, the shadowing component is considered to be uniformly correlated in all directions, allowing predictions to be made by making inferences from spatially-proximate training points. In particular, prior approaches have had success modeling shadowing using a Gaussian-process (GP) [2], [7].

In the case of a fixed base-station (located at  $\vec{b}$ ), we first use the log-fit as a mean function, and then use a standard squared-exponential kernel to specify the expected covariance between signals at two locations,  $x$  and  $x'$ :  $k(x, x') = \sigma_f^2 \exp\{-\frac{\|x-x'\|}{l^2}\}$ . Using this covariance (kernel) function, we can apply standard GP regression techniques to make predictions for a set of sample points [11]:

- 1) **Compute Log-fit:** Fit  $L_0$  and  $a$  using least-squared approach described above.
- 2) **Fit Hyperparameters:** Choose correlation distance  $l$  and function variance  $\sigma_f$  to maximize likelihood of training data.
- 3) **Compute Covariances:** Compute covariance matrix  $K_y$  of training data, and covariance vector  $k_*$  of sample points with respect to training data.  $K_y = K_f + \sigma_d^2 I$ , where each entry  $k_{i,j} \in K_y = k(x_i, x_j)$ .
- 4) **Evaluate prediction:**  $y_{\text{sample}} = k_*^T K_y^{-1} (y_{\text{observed}} - y_{\text{log}})$

These correlative methods have good predictive performance, especially in the case when many training points are available. The main shortcomings of this method are that prediction assumes signals are uniformly correlated in all directions – an assumption which breaks down in the presence of discrete attenuating objects. Direct extension of this method to the case of mobile nodes is also problematic, since training points are in  $\mathbb{R}^4$ , requiring significantly more data to achieve the same performance. Finally, this method is computationally expensive, requiring the inversion of a matrix whose dimension is determined by the number of training points.

#### IV. METHODS

In this section, we extend both the tomographic and correlative methods to the case of multiple mobile nodes. We will describe our modifications to these existing approaches, and

introduce our new approach, MATTE, which also leverages other sensors to infer the location of attenuating objects.

##### A. Correlative Methods for Mobile Nodes

Extending previous approaches of correlative prediction to the case of moving nodes exacerbates the data-sparsity problem. Instead of producing signal predictions for  $\mathbb{R}^2$  (all points in the plane), we now must produce predictions in  $\mathbb{R}^4$  (all possible *pairs* of points in the plane). Since we can't increase the number of signal-strength observations that robots make without slowing down the speed of exploration, this means we have significantly reduced data density. However, we were able to mitigate this problem somewhat by recognizing that our signal-strength models are symmetric with regard to where the transmitter and receiver are – that is, we assume the signal strength is the same from robot A to B as it is from robot B to A. This observation allows us to construct a symmetric distance function,  $d_{s4}$ , which effectively doubles the data density:

$$d_{s4}(\vec{r}_a, \vec{t}_a, \vec{r}_b, \vec{t}_b) = \min \left\{ \begin{array}{l} \|\vec{r}_a - \vec{r}_b\| + \|\vec{t}_a - \vec{t}_b\| \\ \|\vec{r}_a - \vec{t}_b\| + \|\vec{t}_a - \vec{r}_b\| \end{array} \right. \quad (3)$$

In other words,  $d_{s4}$  is a distance metric for pairs of lines that is invariant to rotations of 180 degrees.

In practice, many thousands of observations may be available for use in training the Gaussian process. Due to the  $O(n^3)$  computation cost of matrix inversion, it quickly becomes impractical to include all training data. However, prediction performance is improved as more training points are used. In the case where  $K$  is very sparse, the inversion can be done more quickly, enabling use of more training data. However, the squared exponential kernel is not sparse; two measurements will never have a covariance of exactly zero, even if they are very far apart. By modifying the kernel to have compact support – that is, the kernel is exactly zero once some distance  $\Theta$  is reached – the matrix  $K$  becomes sparse [12]:

$$k_s(x, x') = \max(0, 1 - \frac{\|x - x'\|}{\Theta})^\gamma * k(x, x') \quad (4)$$

The resulting sparsity allows computing  $K^{-1}$  much faster, enabling the use of more training points. Although sparse kernels generally have worse performance, we found in practice that the speed improvement enabled the incorporation of enough extra training points to achieve a net improvement in performance. This sparsification also increases the number of hyper-parameters that must be estimated to a total of 5. In principle, we can learn these additional parameters the same way we learn the parameters for the original covariance function. However, increasing  $\Theta$  will always result in a lower training error and an increased computation due to a less-sparse covariance matrix. To limit worst-case computation time, we do an offline parameter sweep to estimate the best  $\Theta$  and  $\gamma$  subject to a fixed CPU budget.

Together, the symmetric distance metric and the forced sparsification of the correlation function enabled us to adapt existing correlative prediction techniques to the case of

multiple mobile robots, and achieve results competitive with our proposed method. These additions to existing methods form the Multi-robot Gaussian Process “MRGP” method which we include in our evaluation.

### B. Multi-robot Tomography (MRT)

Conceptually, extension of the tomographic methods to the case of multiple robots is relatively straight-forward. The attenuation of each pixel through which a signal passed is estimated in least-squares fashion. The number of pixels,  $p$ , is determined by the grid size such that the pixels completely fill the workspace of the robots. For a grid of width  $w$  and height  $h$ ,  $p = w \times h$  and pixels are labeled  $v_{0,0} \cdots v_{w-1,h-1}$ . In addition, the two path-loss parameters,  $L_0$  and  $a$  are also estimated simultaneously, resulting in  $n = p + 2$  unknowns:  $x = [L_0 \ a \ v_{0,0} \cdots v_{w-1,h-1}]$ . Each of the  $m$  signal strength observations  $y_i$  provide one equation partially constraining a subset of the pixels in addition to the path loss parameters (See Eqns 2 and 1). Together these equations are stacked to form the rows of an  $n \times m$  matrix  $A$ . If  $A$  is full rank,  $x = (A^T A)^{-1} A^T y$ . However, even though  $m$  is generally greater than  $n$ ,  $A$  remains rank deficient, resulting in many possible solutions for  $x$ . Related approaches have solved this by using Tikhonov regularization that enforces smooth changes in attenuation between neighboring pixels [9]. This corresponds to constructing a Tikhonov matrix  $\Gamma$  whose rows correspond to equations of the form  $v_{i,j} - v_{i+1,j} = 0$  and  $v_{i,j} - v_{i,j+1} = 0$  for each  $i, j < w, h$ . The over-constrained solution for  $x$  now takes the form

$$x = (A^T A + \lambda^2 \Gamma^T \Gamma)^{-1} A^T y \quad (5)$$

where  $\lambda$  is a parameter to determine the weight of the smoothness constraints in  $\Gamma$  relative to the observation equations in  $A$ . In contrast to the correlative methods, which scale  $O(m^3)$  with respect to the number of observations, the tomographic methods scale  $O(p^3)$  with respect to the number of pixels. However, unlike the correlative methods,  $A^T A$  is naturally sparse since each pixel is only jointly constrained with a small number of other pixels. This means that sparse matrix-inversion methods, such as a Cholesky decomposition, can perform significantly better than  $O(p^3)$ . In practice,  $A^T A$  is still dense enough (95% zeros) that we are limited to solving grids on the order of  $100 \times 100$  (10000 pixels). For the datasets in our evaluation, this translates to a grid size between 2 and 4 meters. Such large grid sizes poorly approximate the sharp spatial changes in attenuation found in real environments, such as at the border between a building and neighboring free space. Furthermore, in order to avoid over-fitting,  $\lambda$  must be set large enough to allow very little spatial gradient in the attenuation of each pixel (See Fig. 1). In areas where attenuation changes quickly, the predictive power is limited. Our implementation of this approach, Mutli-robot Tomography (MRT), is included in our evaluation.

### C. Multi-sensor ATtenuation Estimation (MATTE)

The approaches we’ve discussed so far focus solely on using signal-strength readings for prediction of future mea-

surements. Many robots already carry additional sensors for mapping and navigation. Intuitively, since the physical structure of an environment (the position of walls and other solid surfaces) influences signal propagation, a map of the environment should help predict signal propagation. This is especially true for sensors like laser range-finders, which tend to have a range of at least 10-30 meters. Incorporating a map does not solve the problem completely though, since the attenuation of a wall depends on the material and LIDAR generally can’t distinguish between a reinforced concrete wall and drywall.

Specifically, we propose the use of occupancy grids derived from laser range-finders to provide a more informed regularization constraint to tomographic methods. The occupancy grids collected by our robots label the world with three classes: known free space, known structure and unknown. This information can provide a much better prior about the attenuating properties of the environment – for example, we generally expect areas which are marked as free space to pass signals with little interference. Similarly, knowing the location of structures can provide a prior about where attenuation should increase dramatically. Besides providing a better prior about the magnitude of attenuation, the occupancy grids also provide a much finer view of the environment. Using the MRT method, we are typically limited to coarse grid sizes (e.g. 3 m for our datasets) due to computational constraints. On the other hand, LIDAR-based occupancy grids can be computed cheaply even for fine-grained grid sizes (we used 10 cm grid in our experiments).

Of the many ways of incorporating this data, we explored explicitly estimating the attenuation of each of the three classes separately. A simple approach to this problem is to fit a single attenuation value to all pixel of the same class. For example, known free space might have an attenuation of  $-0.01dBm$  per meter, whereas structure (e.g. walls) could have an attenuation of  $-0.3dBm$  per meter, and unknown space could be approximated as somewhere in between. This approach is notable in its simplicity – it has no parameters to tune and only 5 degrees of freedom to fit the observed data: two for path loss parameters and three more for the attenuation assigned to each class in the occupancy grid. In general, however, it is a poor assumption that all objects detected by the robots as structure will have the same attenuation. For example, a wooden fence and a brick building have very different effects on a signal, but can appear very similar to a robot’s laser range finder. As our initial experiments confirmed, this model performed poorly in explaining real-world data.

Another way to use the occupancy grids is to better inform regularization methods, for example by enforcing smoothness constraints only between neighboring pixels of the same class. Using a more flexible model is difficult, however, since individually estimating the attenuation for each pixel in the fine-grained occupancy grid results in a poorly constrained, computationally prohibitive problem. The largest map we present has nearly 5 million pixels! Downsampling to a coarser resolution can make the problem

more tractable, but reduces the ability to incorporate features such as doors or walls which may be lost by resampling.

---

**Algorithm 1** MATTE\_UPDATE(trainPoints, map)

---

```

init  $A, b$  from  $\text{bounds}(\text{map})$ 
for all  $p \in \text{trainPoints}$  do
   $r \leftarrow [1, \log_{10}(\|p_r - p_t\|), 0 \dots, 0]$ 
  for all locations  $v_i \in \text{losPath}(p)$  do
     $c \leftarrow \text{map.class}(v_i)$ 
     $r(v_i, c) \leftarrow 1$ 
  end for
   $A \leftarrow \text{appendRow}(r)$ 
   $b \leftarrow \text{appendElement}(p_z)$ 
end for
 $\{A, b\} \leftarrow \text{appendRegularization}(A, b)$ 
 $x \leftarrow \text{cholesky.solve}(A^T A, A^T b)$ 
store  $x$ 
store  $\text{map}$ 
return

```

---



---

**Algorithm 2** MATTE\_PREDICT(testPoints)

---

```

retrieve  $x$ 
 $z \leftarrow \text{vector}(\text{testPoints.len})$ 
for all  $p \in \text{testPoints}$  do
   $z_p \leftarrow x(0) + \log_{10}(\|p\|)$ 
   $z_s \leftarrow \text{attenuationIntegral}(x, p, \text{map})$ 
   $p \leftarrow \text{append}(p, z_p + z_s)$ 
end for
return  $p$ 

```

---

Instead, we propose a method that benefits from the detailed resolution of the map, but has computational complexity comparable to the MRT method. Our technique, called MATTE, separately estimates spatially-varying attenuation for each of the three classes in the occupancy grid. These estimates allow querying any point in the environment and determining, for example, *if* there was structure there, what attenuation it would have. When predicting signal-strength, we first examine pixels in the occupancy grid to determine which class they are, and then perform a look-up on the appropriate attenuation estimate. The benefit of this approach is that we can estimate the spatially-varying attenuation at a resolution independent of the resolution required to adequately map the environment structure. Typically grid sizes of 10 to 20 cm are required to preserve the presence of doors or thin walls – but attenuation rates of a particular material, (e.g. building) tend to vary at a much slower rate. The computational implications of this approach are significant – whereas the resolution of the occupancy grid is typically 10 cm, we have achieved good results with a grid size of 4 m for the spatially varying per-class attenuations, resulting in a reduction by a factor of 1600 in the number of unknowns.

Similar to the previous tomographic approach, we simultaneously estimate the path-loss parameters and the spatially

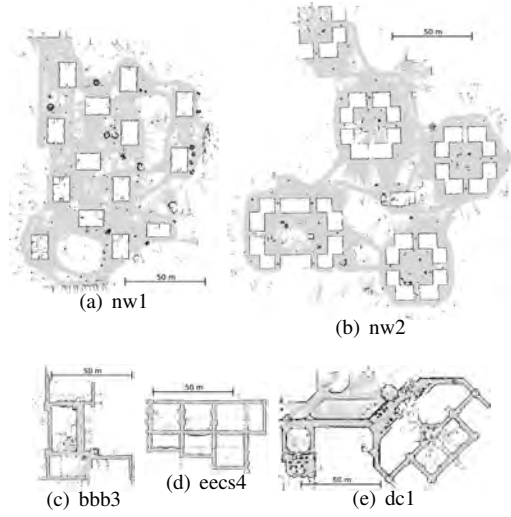


Fig. 2. Maps collected in each of the five datasets we used in our evaluation. From left to right they are: two outdoor sections of the Northwood residential housing complex, the 3rd floor of the Bob and Betty Beyster Building, the 4th floor of the Electrical Engineering and Computer Science Building and the first floor of the Duderstadt Library. White is unexplored, gray is known free-space and black is structure. Each evaluation set consists of back-to-back traversals of three of these environments; meta-parameters are tuned using the remaining two.

varying attenuation. For a workspace  $w \times h$  pixels in dimensions, we estimate a total of  $2 + 3 * w * h$  variables that comprise  $x$ . Although this represents a threefold increase in the number of degrees of freedom over our previous approach,  $A^T A$  is generally more sparse (e.g. 99% sparse in our datasets), ultimately requiring less time to compute a result. An overview of the update and prediction steps are shown in Algorithms 1 and 2. The MATTE algorithm has several important parameters which can either be tuned by hand or estimated automatically from training data. The most important parameter is the relative weight  $\lambda$  given to the smoothing regularizer we described in Eqn. 5. In addition we introduce three parameters to govern the expected attenuation in areas where we have not collected any signal data: a weighting factor  $\phi$  determines the relative weight of this prior, and prior attenuation values in units of  $dBm$  per meter for each class are  $\rho_f, \rho_s, \rho_u$ , for free-space, structure, and unknown space respectively. In practice, we set  $\rho_s = \rho_u$ , allowing free space to have a distinct prior from other areas.

An additional potential advantage of MATTE is that new occupancy grids can be incorporated cheaply, as the underlying spatially varying attenuation does not necessarily need to be updated when the map changes. For example, our approach might encode the knowledge that a structure-class pixels in a particular area tend to have an attenuation of  $X$   $dBm$  per meter. If the map of that area is later expanded, that information will be able to provide an estimate for the attenuation expected there, without needing to recompute  $x$ . This allows our approach to potentially be adapted to run online.

Eval.	Method	Meta-parameters			
1	MATTE MRT	Train: {nw2,bbb3}	Test: {eecs4,dc1,nw1}		
		$\lambda=2.0$	$\phi=.2$	$\rho_f=-.1$	$\rho_{u,s}=-.3$
		$\lambda=0.1375$			
2	MATTE MRT	Train: {eecs4,dc1}	Test: {bbb3,nw2,nw2}		
		$\lambda=1.225$	$\phi=0.05$	$\rho_f=0.2125$	$\rho_{u,s}=-0.2875$
		$\lambda=0.1125$			
3	MATTE MRT	Training: {bbb3,eecs4}	Test: {nw2,nw1,dc1}		
		$\lambda=2.0$	$\phi=0.275$	$\rho_f=0.2875$	$\rho_{u,s}=-0.4875$
		$\lambda=0.1250$			
4	MATTE MRT	Train: {nw1,eecs4}	Test: {nw2,dc1,bbb3}		
		$\lambda=1.38$	$\phi=0.3$	$\rho_f=-.1$	$\rho_{u,s}=-.3$
		$\lambda=0.1125$			
5	MATTE MRT	Train: {dc1,nw1}	Test: {eecs4,bbb3,nw2}		
		$\lambda=2.0$	$\phi=0.1$	$\rho_f=-0.15$	$\rho_{u,s}=-.3$
		$\lambda=0.5875$			

TABLE I

AUTOMATICALLY DETERMINED PARAMETER SETTINGS FOR EACH EVALUATION USING COORDINATE DESCENT ON THE TRAINING SET. Some parameters were fixed for performance reasons, including the grid sizes for MATTE and MRT methods, at 4.0 meters and 3.0 meters respectively. For MRGP, the number of training points was set to  $\min(600, .05 * m)$ ,  $\Theta$  was fixed to 20.0, and  $\gamma$  to 3. Also  $\sigma_d$  and  $\sigma_f$  were fixed for numerical stability reasons at 1.5 and 1.0 respectively. For MRGP, coordinate descent fixed  $l$  to 28.75 on all training data.

## V. EVALUATION

We evaluated the three primary methods we have discussed to determine which methods were most successful at predicting real-world signal-strength measurements. The main purposes of our evaluation is to show that previous signal-strength predictions methods using a fixed base station can be extended to the more general case where all nodes are mobile. We also show that in many cases sensor data can be effectively leveraged to further improve signal-strength predictions. Several of our models have many degrees of freedom, so our evaluation also seeks to show that the methods we present can generalize well from previous observations to the prediction of future signal measurements.

### A. Test Apparatus

We used a robot platform our lab custom designed for urban reconnaissance [13]. We outfitted three of our 14 robots with additional 2.4 GHz TP-Link WiFi radios which were programmed to report signal-strength measurements at 20 Hz. Our robots are equipped with 3D laser range finders, in addition to IMUs and odometers, enabling them to produce high-quality globally consistent maps using Simultaneous Localization and Mapping (SLAM) algorithms [14], [15], [16]. This capability allows us to quickly collect large amounts of signal-strength data that is co-registered with a global grid.

We used these three robots to collect a series of 5 datasets in various indoor and outdoor environments, spanning a total area over 40,000  $m^2$ . The datasets consist of three indoor environments, **bbb3**, **eecs4** and **dc1** for short, as well as two

larger outdoor environments, abbreviated **nw1** and **nw2** (See Fig. 2 for details). Real-world rescue robots must operate in mixed indoor-outdoor urban environments. In order to mimic these conditions, we randomly selected a sequence of three of the environments for testing, guaranteeing a mix of indoor and outdoor datasets. Each of the datasets consist of exploration missions so very few sensor measurements can be considered duplicates, since robots do not tend to retrace their steps. This enables us to explicitly test the predictive performance of each method, rather than their recall abilities.

For each randomly-selected set of three areas, we replayed the signal-strength observations and corresponding maps in two second increments. After each step, the methods were tested on their prediction performance of the next 10 intervals of future signal strength data (e.g from 0-2 seconds, 2-4 seconds, 4-6 seconds, etc.). As the traversal played out, methods had access to an increasing amount of training data, but the amount of testing data was relatively similar at each step. Several of the methods also have meta- or hyper- parameters which need to be tuned before the start of a traversal. We used the two remaining areas not selected for the test set to train meta parameters using a compass search. We include results for 5 such randomly generated traversals. The parameters selected by the compass search for each method on each evaluation are shown in Table I. All of the methods we presented were tuned to use comparable amounts of computational resources. While each method exhibits significantly different asymptotic growth, we set parameters which resulted in roughly equal CPU use over the course of an evaluation set. Run-times for MATTE ranged between 2 seconds to process the bbb3 dataset up to 30 seconds for the significantly larger nw2 dataset. For the GP method, times varied between 15 seconds for bbb3 datasets and 20 seconds on nw2.

### B. Results

The testing results for the five evaluation sets are displayed in Fig. 3. As expected, all methods are better at making short-term rather than long-term predictions. This is a result of the fact that measurements nearer in the future are more likely to be similar to existing measurements, or the fact that attenuating objects impacting observations in the near future are likely to be correlated with sensor data the robots have just now collected.

All of the methods we have implemented show competitive performance, especially for predictions between 0 and 10 seconds in the future. However, MATTE significantly outperforms the other methods in evaluations 1 and 5. In evaluations 2 and 4, it performs comparable to the tomographic method. However, in evaluation 3, our method exhibits worse performance than the other methods. An examination of the meta-parameters determined via compass search show evidence of over-fitting the training set, which by nature of our randomly selected test sets, happened to both be exclusively indoors. This is manifest as a positive attenuation prior for free-space ( $\rho_f$ ), consistent with the wave-guide effect sometimes seen in hallways. In the other evaluation

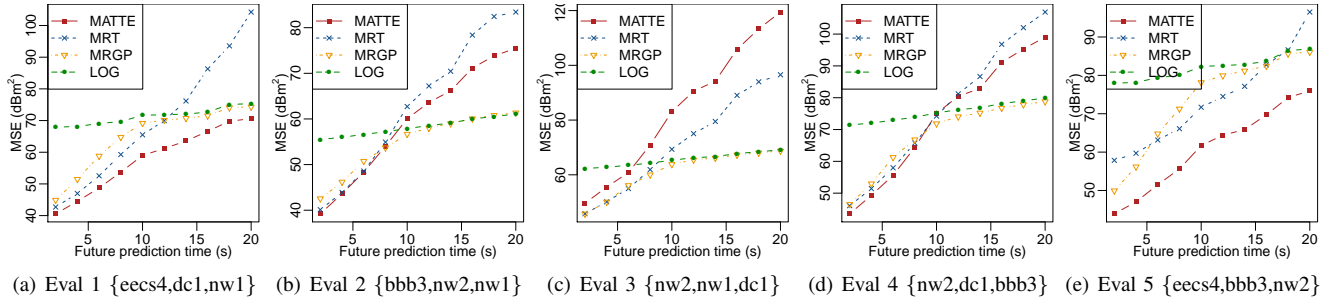


Fig. 3. MSE of prediction accuracy for each method evaluated on 5 randomly chosen sequences over the testing datasets. Error bars are bounded by 0.8 dBm<sup>2</sup> and are omitted for clarity. Our proposed method, MATTE, is compared to extensions of previous methods MRGP and MRT. Our method exploits generally performs better for near-term predictions where the robots’ sensor data provides an informative prior. Poor performance of MATTE in evaluation 3 is attributable to the randomly generated training set which contains only indoor datasets, while the testing set contains both indoor and outdoor sets.

sets, there was at least one outdoor dataset used for training, which helped to mitigate this type of over fitting. In most of the evaluation sets, we see that MATTE can out-perform the correlative method for short-term predictions up to ten seconds in the future. For longer-term predictions, where training data is mostly useless, it is difficult to beat the log baseline.

## VI. CONCLUSION

In this work, we have explored the difficult problem of predicting signal strength when all nodes are actively exploring a variety of real world environments. This problem is challenging because robots typically only sample the possible signal propagation paths very sparsely, making generalization beyond a naive log-fit difficult. We extended several existing signal strength prediction methods to the case of multiple robots conducting exploration-style missions in mixed indoor-outdoor urban environments. We additionally suggested a new method for signal prediction which uses additional sensors already in use by many autonomous robots to better estimate the regions of attenuation in an environment. Our methods performs competitively with the other approaches, and in some cases performs much better. Key to our approach is the ability to estimate the attenuation properties of the environment at a coarse level, but still use a fine-grained spatial model of an environment derived from additional sensors. It is interesting that both correlative and tomographic methods exhibit such similar performance, as they employ very different approaches to the same problem. This suggests that both methods are exploiting similar aspects of signal strength propagation.

Our work has important applications to autonomous robot teams which are collaborating to achieve a joint goal. By introducing more advanced signal-strength modeling techniques, such teams can better predict when robots can expect to communicate, allowing them to plan their future actions

by explicitly including communication as a constraint.

In the future we look to continue exploring methods for effectively predicting signal strength. In particular, it may be possible to leverage additional sensing modalities to improve prediction. Incorporating this predictor in an online planning system would also serve to further validate our approach.

## REFERENCES

- [1] M. Rooker and A. Birk, “Multi robot exploration under the constraints of wireless networking,” *Control Engineering Practise*, vol. 15, no. 4, pp. 435–445, 2007.
- [2] N. Michael, M. Zavlanos, V. Kumar, and G. Pappas, “Maintaining connectivity in mobile robot networks,” *Experimental Robotics*, 2009.
- [3] G. Hollinger and S. Singh, “Multi-robot coordination with periodic connectivity,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4457–4462.
- [4] A. Ghaffarkhah and Y. Mostofi, “Channel learning and communication-aware motion planning in mobile networks,” in *American Control Conference (ACC)*, July 2010, pp. 5413–5420.
- [5] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, “Coordinated multi-robot exploration,” *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, june 2005.
- [6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [7] M. Malmirchegini and Y. Mostofi, “On the spatial predictability of communication channels,” *Wireless Communications, IEEE Transactions on*, vol. 11, no. 3, pp. 964–978, march 2012.
- [8] J. Fink and V. Kumar, “Online methods for radio signal mapping with mobile robots,” in *International Conference on Robotics and Automation*, 2010.
- [9] J. Wilson and N. Patwari, “Radio tomographic imaging with wireless networks,” *Mobile Computing, IEEE Transactions on*, vol. 9, no. 5, pp. 621–632, May 2010.
- [10] J. Wilson, N. Patwari, and F. Vasquez, “Regularization methods for radio tomographic imaging,” in *2009 Virginia Tech Symposium on Wireless Personal Communications*, 2009.
- [11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2005.
- [12] H. Zhang, M. Genton, and P. Liu, “Compactly supported radial basis function kernels,” Tech. Rep., 2004.
- [13] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goedel, M. Bulic, J. Crossman, and B. Marinier, “Progress towards multi-robot reconnaissance and the MAGIC 2010 competition,” *Journal of Field Robotics*, 2012.
- [14] J. Strom and E. Olson, “Occupancy grid rasterization in large environments for teams of robots,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [15] E. Olson, “Real-time correlative scan matching,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, June 2009.
- [16] F. Dellaert, “Square root SAM,” in *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, USA, June 2005.



# IPJC: The Incremental Posterior Joint Compatibility Test for Fast Feature Cloud Matching

Edwin B. Olson<sup>1</sup> and Yangming Li<sup>2</sup>

**Abstract**—One of the fundamental challenges in robotics is data-association: determining which sensor observations correspond to the same physical object. A common approach is to consider groups of observations simultaneously: a constellation of observations can be significantly less ambiguous than the observations considered individually. The Joint Compatibility Branch and Bound (JCBB) test is the gold standard method for these data association problems. But its computational complexity and its sensitivity to non-linearities limit its practical usefulness.

We propose the Incremental Posterior Joint Compatibility (IPJC) test. While equivalent to JCBB on linear problems, it is significantly more accurate on non-linear problems. When used for feature-cloud matching (an important special case), IPJC is also dramatically faster than JCBB. We demonstrate the advantages of IPJC over JCBB and other commonly-used methods on both synthetic and real-world datasets.

**Index Terms**—Data association, joint compatibility test, SLAM

## I. INTRODUCTION

Data association is the problem of determining which observations correspond to the same object. It is at the core of the Simultaneous Localization and Mapping (SLAM) problem and visual navigation: it is only by re-observing a landmark that a map becomes over-constrained and therefore more robust to the errors associated with any single observation.

SLAM systems are often described in terms of the two “halves” of the problem: the front-end performs the sensor processing and data association, while the back-end computes the maximum-likelihood map subject to the observations and data-associations output by the front-end. In recent years, the raw computational performance of back-ends has increased dramatically: maps with millions of landmarks and observations can be optimized [1].

While back-end systems are now very fast, the quality of their output is entirely dependent on the accuracy of the front-end. In particular, an incorrect data association (in which the front-end erroneously asserts that two physically-distinct landmarks are in fact the same landmark) forces the back-end to distort the map to bring those two landmarks closer together. Even a single data-association error can lead to divergence of the entire map.

Consequently, the quality of a front-end system has an enormous impact on the quality of the resulting map. Too many loop closures (i.e., false positives) lead to catastrophic

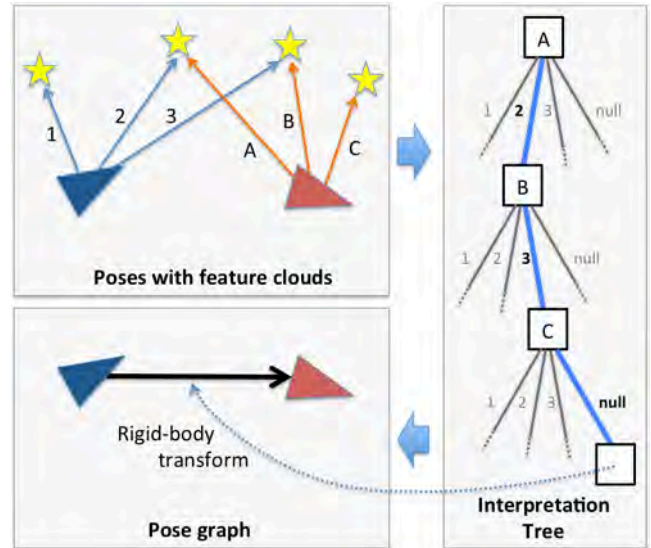


Fig. 1. IPJC Overview. A robot observes “feature clouds” from two different poses (top left), and IPJC matches them by searching a tree and computing a compatibility cost. The rigid-body transformation that results from the matching process can be used in a pose-graph SLAM formulation. IPJC is similar to JCBB, but when applied to feature cloud matching, produces better results in less time.

failures, while too few loop closures (false negatives) lead to a less-constrained map of lower overall quality.

A common approach to improving the quality of data-association systems is to consider multiple observations as a set. A reasonable analogy is that it is difficult to recognize a star given an image of it, but recognizing a constellation is much easier and less error-prone. When matching groups of features, it is critical to consider the correlations between measurements. In general, the set of observations will not match perfectly with the prior estimates of the landmark locations. Due to the correlations between these observations, some misalignments are more likely than others. For example, suppose an image of a constellation of stars is taken. The individual positions of the stars in the image are highly correlated: they all depend on where the camera was pointing. If all of the stars appeared to be shifted uniformly with respect to their a priori estimated positions, the errors could be easily explained in terms of a camera pointing error. On the other hand, if the stars were shifted randomly with respect to their a priori estimated positions, one might instead conclude that the image is of a different set of stars. In other words, proper consideration of the correlations between observations can have a significant effect on the data

<sup>1</sup> Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 ebolson@umich.edu

<sup>2</sup> Institute of Intelligence Machines, Chinese Academy of Sciences, Hefei, Anhui, 230031 ymli@iim.ac.cn

association process.

The gold standard method is the Joint Compatibility Branch and Bound (JCBB) test [2], which searches for the largest set of data associations subject to a bound on the  $\chi^2$  error. Conceptually, JCBB builds an “interpretation tree”; at each level of the tree, an observation is associated with one of the landmarks in the map (or to a “null” hypothesis representing the possibility that no landmark matches the observation.) A path from the root of the tree to any node encodes a set of data-associations, and JCBB explicitly computes a cost related to the probability of that set of data-associations.

The computational cost of JCBB can be substantial; at every node in the tree, the joint compatibility must be computed. Even when computed in a clever incremental fashion, this involves an operation of cost  $O(m^2)$  at level  $m$  in the tree. This cost can be prohibitive in some applications and has led to a number of alternative approaches.

Several authors have suggested faster but less probabilistically-motivated methods for validating data-association hypotheses. One trend has been to implicitly identify groups of compatible hypotheses by considering their pairwise-consistency; this can be viewed as a max-clique [3] or spectral graph partitioning [4] problem. Finding loop closures, a common task in pose-based SLAM, is an application of data-association. A recurring idea is to look for sequences of loop closures that form a closed topological loop: the composition of the loop closures in the loop should approximately be the identity matrix [5], [6], [7].

In addition to high computational cost, JCBB is sensitive to non-linearities. JCBB estimates the joint compatibility by linearizing around the prior and considering the entire set of data associations as a single large linear update. However, more accurate results could generally be obtained by actually computing the posterior after each observation; this results in a better estimate of the posterior and will generally improve the linearization point used to estimate the compatibility of successive data association pairings.

Many data association algorithms target the case where the location of the landmarks is explicitly estimated—i.e., where the state vector is enlarged to contain the position of each landmark. Alternatively, as seen in pose-based SLAM algorithms and in many camera-based applications, the landmarks are *not* added to the state vector. Instead, the motion between two poses is found by matching the feature observations between those two poses. Each pose records a cloud of features observed from that pose, and the problem becomes one of “feature cloud” matching (see Fig. 1). This approach is often used when each pose observes a large number of landmarks: adding all of these landmarks to the state vector can quickly tax even very fast back-end systems. Systems using cameras or 3D LIDAR sensors can extract hundreds of features from a single robot pose, for example.

In this paper, we propose a new data association method that, like JCBB, is probabilistically rigorous. However, it provides better accuracy in non-linear settings and, in the

case of feature cloud matching, dramatic runtime speedups. The contributions of this paper are:

- We propose a posterior-based data association test, motivate it in terms of the  $\chi^2$  of a least-squares optimization, and show that it is equivalent to JCBB. This algorithm, Posterior Joint Compatibility (PJC), serves as the basis for the remainder of our algorithms.
- We propose the Incremental Posterior Joint Compatibility (IPJC) test, which exploits the probabilistic structure of feature cloud matching. This method is both more accurate in non-linear settings and dramatically faster than JCBB. We further show how to accelerate the process further, leading to the IPJC-Fast algorithm.
- We demonstrate our proposed methods along side JCBB, RANSAC, and SCNN on a range of synthetic and real-world problems. This data supports our claim that IPJC and IPJC-Fast out-perform other methods.

## II. A REVIEW OF JCBB

Our method is similar in most respects to JCBB [2]: given a set of  $m$  observations of  $n$  features, we search an “interpretation tree”. This tree has  $m$  levels, and at each level of the tree, we consider  $n + 1$  possible data-associations for the  $m^{th}$  observation. (Each observation could match any of the  $n$  landmarks, or could match none of them.) A path from the root to any node represents a set of data-associations. Our goal is to find a “good” data-association for every observation.

How is the “goodness” of a set of data-associations evaluated? Given a state estimate  $x$  with covariance  $P$ , we use a domain-specific sensor model (assumed known) to compute predicted observations  $\hat{z}$ . Given the matrix  $H$  of partial derivatives of  $x$  with respect to  $\hat{z}$ , the uncertainty of the predicted observations due to our uncertainty of  $x$  is simply  $HPH^T$ .

We assume that our actual observations  $z$  are contaminated by noise with covariance  $V$ , and that the matrix of partial derivatives of the noise variables with respect to  $z$  is  $G$ . The uncertainty of the actual observations due to this underlying sensor noise is simply  $GVG^T$ . The combined uncertainty of our prior and observation is given by:

$$C = HPH^T + GVG^T \quad (1)$$

The discrepancy between our actual and predicted observations is  $e = z - \hat{z}$ . We can now write the cost function used by JCBB as a Mahalanobis distance:

$$\chi^2 = e^T C^{-1} e \quad (2)$$

In principle, we might wish to identify the maximum likelihood set of data-associations (or equivalently, the set of data associations with the minimum Mahalanobis distance). To compute this, we would need to know the likelihood of an observation not matching any landmark—a quantity typically not known. Instead, JCBB searches for the largest set of non-null data-associations such that the Mahalanobis distance is less than a threshold. This threshold is typically



expressed in terms of the  $\chi^2$  distribution for an appropriate number of degrees of freedom.

A naive approach would be to consider each leaf of the tree, compute its Mahalanobis distance, and select the best. However, the interpretation tree is quite large: it has  $(n + 1)^m$  leaves. Fortunately, the search space can be pruned by employing the branch-and-bound method. The key idea is that the Mahalanobis distance can be computed for any partial set of data-associations. Since the Mahalanobis distance can only increase with additional data associations, this serves as an admissible lower-bound for all of the node's children. Consequently, we can prune any sub-tree that could not be better than the best-known solution. A more careful description of JCBB can be found in [2].

The major computational cost in JCBB is to the cubic cost of inverting the  $C$  matrix in Eqn. 2. As pointed out by the original JCBB paper, the inverse of  $C$  at level  $m$  can be computed in terms of the inverse of  $C$  at level  $m-1$ , reducing the complexity to quadratic. Even with this optimization, the cost of repeatedly evaluating the Mahalanobis distance quickly becomes a bottleneck— particularly if there are a large number of observations.

### III. PROPOSED METHOD

#### A. Posterior Joint Compatibility

We begin by deriving a different way of writing the Mahalanobis cost function used by JCBB. Given a putative set of data-associations, suppose we compute the posterior value of  $x$ , which we denote as  $x^+$ . (For clarity, we will denote the prior value of  $x$  as  $x^-$ .) This posterior can be computed in a variety of ways, including Extended Kalman Filtering [8], non-linear optimization [9], etc. In this work, we use the Iterated Extended Kalman Filter [10].

Suppose that we update the state estimate using the Extended Kalman Filter as follows:

$$C = HPH^T + GVG^T \quad (3)$$

$$K = PH^T C^{-1} \quad (4)$$

$$e = z - \hat{z} \quad (5)$$

$$d = Ke \quad (6)$$

$$x^+ = x^- + d \quad (7)$$

We can compute the  $\chi^2$  of the posterior as the sum of the  $\chi^2$  of the observations and the prior evaluated at  $x^+$ . Note that the posterior residual for the observations is not  $e$  (because  $\hat{z}$  reflects the prior estimate of  $z$ ); due to the change in our state estimate, the posterior observation residual becomes:  $z - (\hat{z} + Hd) = e - Hd$ .

In summary, the posterior  $\chi^2$  can be written as:

$$\chi^2 = (e - Hd)^T (GVG^T)^{-1} (e - Hd) + d^T P^{-1} d \quad (8)$$

We now show that this expression is equivalent to the one used by JCBB. To see this, we begin by substituting  $d = Ke$ :

$$\chi^2 = (e - HKe)^T (GVG^T)^{-1} (e - HKe) + (Ke)^T P^{-1} (Ke) \quad (9)$$

And now factoring out  $e^T$  to the left and  $e$  to the right:

$$\chi^2 = e^T [(I - HK)^T (GVG^T)^{-1} (I - HK) + K^T P^{-1} K] e \quad (10)$$

We'll now consider the cost function used by JCBB, showing that it can be simplified to the same expression as Eqn. 10. Recall that the Mahalanobis distance used by JCBB can be written as  $\chi^2 = e^T C^{-1} e$ . Let us begin by focusing on the inner term  $C^{-1}$ .

$$C^{-1} \quad (11)$$

$$[I + K^T H^T - (HK)^T] C^{-1} \quad (12)$$

$$[K^T H^T + (I - HK)^T (GVG^T)^{-1} (GVG^T)] C^{-1} \quad (13)$$

$$[K^T H^T + (I - HK)^T (GVG^T)^{-1} (C - HPH^T)] C^{-1} \quad (14)$$

$$[K^T (P^{-1} PH^T C^{-1} C) + (I - HK)^T (GVG^T)^{-1} (I - HK) C] C^{-1} \quad (15)$$

$$[K^T P^{-1} KC + (I - HK)^T (GVG^T)^{-1} (I - HK) C] C^{-1} \quad (16)$$

$$K^T P^{-1} K + (I - HK)^T (GVG^T)^{-1} (I - HK) \quad (17)$$

In Eqn. 12, note that  $(HK)^T = K^T H^T$ . In Eqn. 13, we multiply by  $GVG^T$  and its inverse; note also that  $I - (HK)^T = (I - HK)^T$ . In Eqn. 14, we substitute  $GVG^T = C - HPH^T$ , which follows from Eqn. 3. In Eqn. 15, we use  $P^{-1}P = I$  and  $C^{-1}C = I$ , and we factor  $(C - HPH^T)$  as  $(I - HK)C$ . In Eqn. 16, we substitute  $K = PH^T C^{-1}$ . Finally, in Eqn. 17, we distribute the  $C^{-1}$  factor.

We can now write the cost function used by JCBB in terms of this final expression for  $C^{-1}$ :

$$\chi^2 = e^T [(I - HK)^T (GVG^T)^{-1} (I - HK) + K^T P^{-1} K] e \quad (18)$$

Eqn. 10 and Eqn. 18 are identical; thus, both formulations compute the same value. In some ways, the posterior-based test is more intuitive, since it corresponds to an explicit minimization of the same metric function used by non-linear SLAM systems.

#### B. Feature Cloud Matching

In a standard landmark-based SLAM system, the position of each landmark is added to the state vector, and each observation of that landmark improves the estimate of its position. The position of the landmarks become correlated, due to the fact that different sets of landmarks are observed at different points in time.

In contrast, a feature-cloud matching approach does not add landmarks to the state vector, and thus does not attempt to compute optimal estimates of their positions. Instead, landmark detections from two poses  $A$  and  $B$  are used to estimate the motion of the robot between  $A$  and  $B$ .

Perhaps the most canonical example of feature-cloud matching is scan-matching: two scans are aligned in order to recover the motion of the robot, but the scans do not update a global model of the underlying structure that led to the observations. Iterative Closest Point (ICP) is often used

for both 2D and 3D [11]. However, ICP methods require good initial estimates, which are not always available. In this case, features can be extracted from the data and the features are explicitly associated with each other. The relationship between the two poses can then be computed from the feature correspondence. Feature-cloud matching examples include matching camera data for navigation [12], and computing a rigid-body transformation to align two 3D LIDAR point clouds [13]. The latter case demonstrates how the same feature-cloud matching process applies to object recognition (matching an observation to a model).

A feature-cloud matching system is generally a good choice when many landmarks are detected at every pose. First, it can be impractical to add them all to the state vector: adding hundreds of landmarks at every robot pose would quickly bog down even the fastest SLAM implementations. Second, it is often unnecessary for accurate mapping: when many landmarks are detected simultaneously, the rigid-body transformation relating the two poses tends to be highly over-constrained, which greatly reduces the impact of noise in individual observations.

Feature-cloud matching can be performed using JCBB, but it is expensive to do so. First, the quadratically-increasing cost of incrementally computing the Mahalanobis distance at each level in the interpretation tree quickly becomes a bottleneck. Second, the depth of the interpretation tree is equal to the number of observations, which can measure in the hundreds.

### C. Incremental Posterior Joint Compatibility

The posterior joint compatibility test suggests an alternative approach for performing data association on feature clouds. The basic idea is to exploit the fact that feature observations are conditionally independent given the rigid-body transformation that relates poses  $A$  and  $B$ . In other words, the critical quantity that needs to be estimated is the rigid-body transformation  $T$  that projects points from coordinate frame  $B$  into coordinate frame  $A$ . Everything else needed to compute the posterior joint compatibility can be recovered once  $T$  is known.

Our approach is summarized below. For clarity, we provide example matrix and vector dimensions assuming that a robot is operating in the plane, i.e., that rigid-body transformations have three degrees of freedom and that landmarks are 2D point features; however, our method is not limited to this case.

- 1) Assume a prior on  $T$ , or alternatively, descend sufficiently far down the interpretation tree such that an initial solution to  $T$  can be computed. The state vector is  $3 \times 1$  and the covariance matrix is  $3 \times 3$ .
- 2) At level  $i$  of the interpretation tree:
  - a) Initialize a landmark based on observation  $i$  from pose  $A$ . This enlarges the state vector to  $5 \times 1$  and the covariance to  $5 \times 5$ . Because this observation was made from position  $A$ , it does not depend on  $T$ . Consequently, the covariance matrix will be block diagonal.

- b) When associating observation  $i$  from pose  $A$  with observation  $j$  from pose  $B$ , perform an EKF-like update. This will update the posterior value of  $T$  and the landmark location. Because the observation model is a function of both  $T$  and the landmark position, the covariance matrix becomes dense.
- c) We do not need to maintain a full state estimate over  $T$  and the landmark position, so we now marginalize out the landmark position. The state vector is now reduced to its original size of  $3 \times 1$ .

This method incrementally computes the posterior rigid-body transformation  $T$  as we traverse the interpretation tree. Critically, the computational time required at each node in the tree is *constant*, as opposed to *quadratically increasing* with JCBB<sup>1</sup>.

An advantage of this approach is that the posterior is improving in quality as we travel down the interpretation tree. This means that if the initial estimate of  $T$  was poor, JCBB might encounter significant error due to linearization effects. Because JCBB does not update the prior as it traverses the interpretation tree, this error will affect every computation in the tree. With the approach above, the quality of the estimate improves as we traverse the tree, decreasing the effect of linearization errors.

However, in order to use this method in a branch-and-bound type of search, we need to be able to compute the posterior  $\chi^2$  at each level of the tree. This might seem to be problematic, since we have marginalized-out the posterior positions of the landmarks. However, as we've shown previously, the desired  $\chi^2$  error can be computed as a sum of the  $\chi^2$  of both individual sets of observations with respect to the posterior. In other words, the posterior  $\chi^2$  can be computed in terms of the posterior  $T$  using the following procedure:

- 1) Compute the  $\chi^2$  error associated with the prior on  $T$  (characterized by mean  $\mu_T$  and covariance  $\Sigma_T$ ),  $\chi_T^2 = (x - \mu_T)^T \Sigma_T^{-1} (x - \mu_T)$ .
- 2) For each associated pair of observations  $i$  and  $j$ 
  - a) Compute the posterior position of the landmark given the two observations and  $T$ . We do this by projecting observation  $j$  into coordinate frame  $A$  using rigid-body transformation  $T$ . Note that for the purposes of this projection,  $T$  has no uncertainty: it is already the desired posterior transformation.
  - b) Combine the uncertain observation  $i$  and the projected uncertain observation  $j$  using an EKF update-like step.
  - c) Compute the  $\chi^2$  of both observations with respect to this posterior, and add it to the total.
- 3) Return the sum of all  $\chi^2$  terms computed.

Note that each time we wish to compute the  $\chi^2$  error for the set of data associations, we re-compute the posterior po-

<sup>1</sup>The computational complexity can be further reduced by modifying the EKF update step so that the values discarded during marginalization are never actually computed.

sitions for each landmark. This is because the posterior value of  $T$  is updated at each level of the tree, and this affects the  $\chi^2$  error associated with each of the landmark pairs. If we had not marginalized out the landmark positions at each level of the tree, this re-computation would not be necessary. Despite the seeming inefficiency of this procedure, we have traded the quadratic cost of maintaining the landmark posteriors for a *linear* cost associated with the algorithm above.

In summary, the IPJC algorithm follows the same general pattern as JCBB; an interpretation tree is constructed, and a search is conducted to find the best set of data associations. The principle difference is that the  $\chi^2$  error is computed in terms of the posterior, rather than the prior. On linear problems, IPJC computes *exactly* the same result, but IPJC produces better quality results on non-linear problems due to the steady improvement in the quality of the posterior. Even more usefully, we show that by formulating the  $\chi^2$  in terms of the posterior, the complexity of the computations performed at each node can be reduced from quadratic asymptotic complexity to linear.

#### D. IPJC-Fast

The dominant computational cost of IPJC results from constantly recomputing the posterior positions of the landmarks and the resulting  $\chi^2$  scores. We now describe an improvement to IPJC that dramatically decreases computational complexity without compromising the accuracy of the method.

The essential observation is that the  $\chi^2$  strictly increases as we progress down the interpretation tree. (This follows from the fact that, at each level of the tree, we compute the state estimate with the minimum  $\chi^2$  error. Any future modifications to the state estimate can only increase the error.)

During the first few data associations, the state estimate for  $T$  often changes significantly. But as we traverse farther down the interpretation tree,  $T$  becomes more confident and the state changes grow smaller. As a result, the  $\chi^2$  for earlier observations do not change very much.

This suggests a simple strategy: instead of recomputing the  $\chi^2$  cost of each observation at every node in the tree, simply cache the values from the previous level. The resulting  $\chi^2$  estimate will be strictly smaller than the true  $\chi^2$ . If this lower-bound of the  $\chi^2$  cost is greater than the  $\chi^2$  threshold used in the branch-and-bound search, the sub-tree rooted at that node can be pruned.

Recall also that the branch-and-bound search also prunes nodes whose  $\chi^2$  error is worse than the best-known solution. Whenever a node appears to be the new “best” solution, we need to recompute the correct  $\chi^2$  error in order to ensure that it is correct.

As our results demonstrate, this “lazy” strategy pays off: this algorithm, which we call IPJC-Fast, produces exactly the same results as IPJC, but does so in a fraction of the time. Neglecting the occasional need to recompute the full  $\chi^2$  cost, the asymptotic complexity at each node in the interpretation

tree is now  $O(1)$ . This is in comparison to the quadratic costs of JCBB.

## IV. RESULTS

### A. Simulation Results

In this experiment, and the other synthetic experiments that follow, we simulated a planar robot operating in a field of randomly-placed landmarks. In the linear experiments, the robot has no orientation (or equivalently, has a “perfect” compass) and observes the distance in the  $\hat{x}$  and  $\hat{y}$  directions to landmarks. In the non-linear experiments, the robot acquires range-bearing observations. The sensor range and obstacle density are configured so that around 15 landmarks are visible from any given position. The robot trajectory is sampled such that, between adjacent poses, there are around 10 matching landmarks.

### B. Accuracy in comparison to the ideal $\chi^2$ distribution

We now wish to demonstrate that IPJC computes better estimates of the true compatibility cost in non-linear problems. Our methodology relies on synthetic datasets in which the magnitude of noise and data association can be known with certainty. In this case, the compatibility cost of the *true* data associations should obey a  $\chi^2$  distribution.

Fig. 2 shows histograms of the computed compatibility cost on a large number of trials versus the ideal distribution (given by a  $\chi^2$  distribution of the appropriate parameters). Results are shown for JCBB, PJC, and IPJC, for both low-noise and high-noise situations. All algorithms produce reasonable results in low-noise situations, which is sensible since linearization effects are minimized when noise is low.

However, in high-noise situations, the performance of the algorithms is radically different. IPJC’s compatibility costs follow the correct distribution much more closely than either JCBB or PJC. (Recall that PJC does not incrementally update the posterior, and so does not have the robustness to noise that IPJC has.)

We can quantify the similarity of the histograms to the ideal  $\chi^2$  distributions by computing the likelihood of sampling the empirical distribution from the ideal distribution. These likelihoods substantiate our claims: IPJC’s log-likelihood is -110.9, whereas JCBB’s log-likelihood is -5748.6.

It is clear from Fig. 2 that JCBB has a tendency to over-estimate the compatibility cost of true data associations due to the effects of non-linearities. This effect is further demonstrated by Fig. 3, which plots the compatibility costs computed by our method versus JCBB. The high compatibility cost peaks computed by JCBB exceed the  $\chi^2$  threshold and result in false negative data associations. In high-noise settings, JCBB incorrectly rejects many of the correct data associations.

### C. False vs True Positives

False negatives are problematic since they deprive a SLAM solution of information that could improve the quality of the map. However, false positives can be catastrophic,

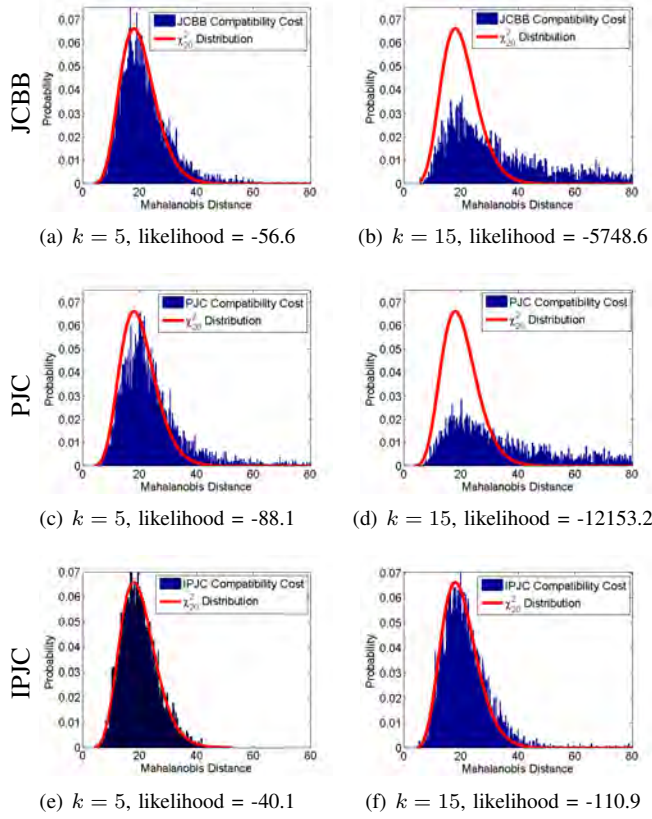


Fig. 2. Compatibility costs versus the ideal distribution. Given ground truth, it is possible to determine the distribution of compatibility costs that a data association algorithm *should* compute; this ideal distribution is shown as a red line. For each algorithm we plot the empirical distribution for two noise levels (algorithms span across rows; noise levels span columns). At low noise levels (left column), each algorithm does fairly well. However, at higher noise levels (right column), IPJC is dramatically more accurate. The log likelihood is shown in each sub-caption and quantifies the similarity between the empirical and ideal distribution; numbers closer to zero represent greater similarity.

leading to divergence. We show the false positive and true positive rates in Fig. 4. The performance of the algorithms is plotted as a function of the noise magnitude, which increases in the  $x$  axis. The figure demonstrates that IPJC and IPJC-Fast produce higher fidelity results: both lower false positive rates and higher true positive rates. Users can trade-off performance in these categories by adjusting the  $\chi^2$  data association threshold.

We have included Sequential Compatibility Nearest Neighbor (SCNN), which greedily matches features one at a time (see [2] for more details). We have also included RANSAC [14] for comparison. For RANSAC, we report results for the consensus threshold that maximized RANSAC’s performance, and for two different iteration limits: one which maximized accuracy (15000), and a second that represents a reasonable compromise between quality and speed (5000).

#### D. Computational Cost

We now consider the computational cost of our methods versus other data association methods. Fig. 5 demonstrates that IPJC-Fast is consistently faster than JCBB, and that it

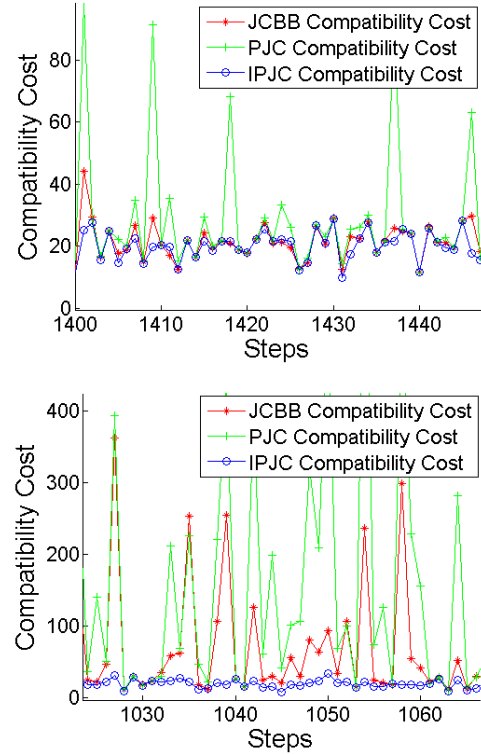


Fig. 3. Comparison of joint compatibility costs. Correct hypotheses are used to calculate the compatibility costs in the proposed methods and JCBB for two different noise levels. Linearization effects cause JCBB and PJC to dramatically over-estimate the compatibility cost, which ultimately causes errors in data association. IPJC computes lower and more accurate compatibility costs.

does not suffer from the spikes in computational complexity arising from linearization error that affect JCBB.

SCNN, as one of the simplest possible data association algorithms, is the fastest method. However, it produces significantly inferior data associations.

The time complexity of JCBB and IPJC-Fast are dependent on the noise level in the problem: as noise increases, more data association hypotheses appear plausible. As a result, more nodes in the interpretation tree must be expanded. As shown by Fig. 6, IPJC and IPJC-Fast are both faster in absolute terms, and exhibit slower growth in time. As expected, SCNN and RANSAC are unaffected by the noise level.

#### E. Victoria Park

Finally, we demonstrate our algorithm on a real-world dataset: Victoria Park. We use the standard tree detection method described in [15] for landmark observations. We established ground truth based on the minimum-error configuration using manually-verified data associations.

In order to make the fairest possible comparison to RANSAC, we tuned the RANSAC parameters to optimize its performance. Beginning with a very large number of iterations, we searched for the consensus threshold that minimized the error in the map, arriving at a value of

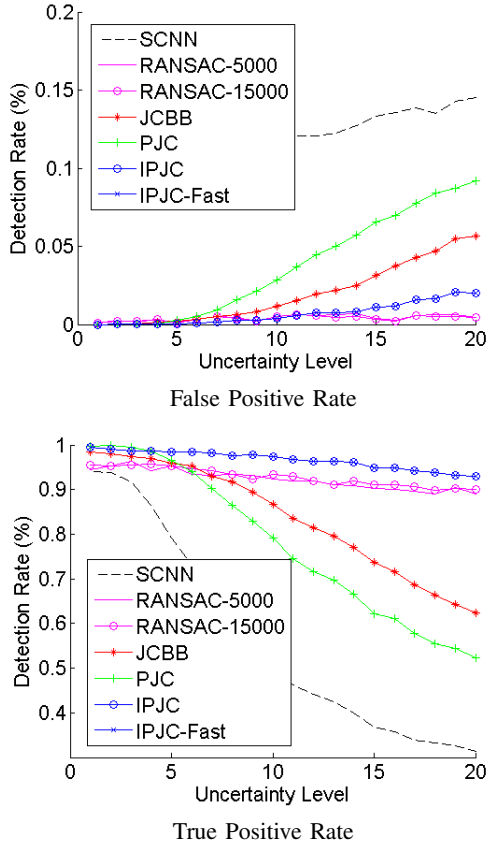


Fig. 4. False and true positive rates. False positives can cause catastrophic errors, but a high true positive rate is necessary to produce accurate maps. For a given  $\chi^2$  threshold, IPJC generates fewer false positives (excluding RANSAC) and more true positives than the other methods. Note that IPJC and IPJC-Fast produce precisely the same data. While RANSAC produces fewer false positives at high noise levels, this comes at the cost of much higher computational costs.

1.0 m. We then reduced the number of iterations in order to improve runtime until the quality of the map started increasing rapidly. In this way, we arrived at 5000 RANSAC iterations, which is sensible given that each pose observes around 15 landmarks, and two associations are required to compute a rigid-body transformation.

#### F. Victoria Results

We built maps using several different data association algorithms and then compute the posterior map using a sparse Cholesky factorization method [16].

TABLE I

VICTORIA PARK MAP ACCURACY. The mean squared error is computed versus a hand-annotated ground truth. IPJC and IPJC-Fast, which produce the same results, are the most accurate of the tested methods.

	SCNN	RANSAC	JCBB	PJC	IPJC	IPJC-Fast
X	Diverged	1.0470	0.5334	0.5731	0.2159	0.2159
Y	Diverged	2.0367	0.9017	0.9934	0.5801	0.5801
$\theta$	Diverged	0.0074	0.0002	0.0001	0.0001	0.0001

The resulting maps are shown in Fig. 7. SCNN makes data association errors that cause the maps to diverge. The

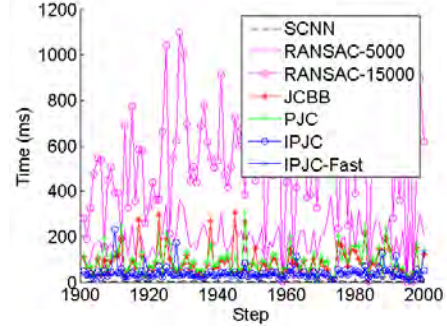


Fig. 5. Computational complexity. Spikes occur when large numbers of features are detected. SCNN is the fastest method, but its accuracy makes it unusable in most problems. IPJC-Fast consistently outperforms JCBB and RANSAC.

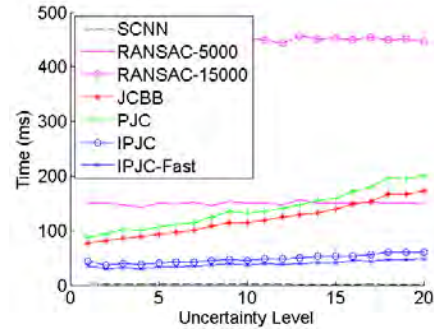


Fig. 6. Computational complexity versus noise level. Joint compatibility methods (including IPJC) tend to require more computation in high noise environments, since more hypotheses appear plausible. However, the growth rate for IPJC-Fast is much lower than for JCBB. RANSAC's complexity is independent of the level of noise, as expected.

remaining methods, RANSAC, JCBB, PJC, IPJC, and IPJC-Fast produce visually indistinguishable maps. However, the maps are not identical: due to differing true and false positive rates, the quality of the maps varies between methods. Table I shows the mean squared error for the various methods. IPJC-Fast (which produces the same results as IPJC, just faster) produces a higher-quality result than the other methods.

These performance differences can be explained by the true and false positive rates; see Table II. IPJC and IPJC-Fast have the highest true positive rate and the lowest false positive rate of any of the methods considered.

We also show the computational costs associated with the different data association methods in Fig. 8. Naturally, SCNN is the fastest (though its quality is poor); IPJC-Fast outperforms all other methods.

TABLE II

VICTORIA PARK TRUE/FALSE POSITIVE RATES. On this real-world data, IPJC and IPJC-Fast outperform all other methods in both true positives and false positives.

	SCNN	RANSAC	JCBB	PJC	IPJC	IPJC-Fast
True pos.	0.2410	0.9132	0.9565	0.9623	0.9818	0.9818
False pos.	0.0608	0.0121	0.0044	0.0018	0.0004	0.0004



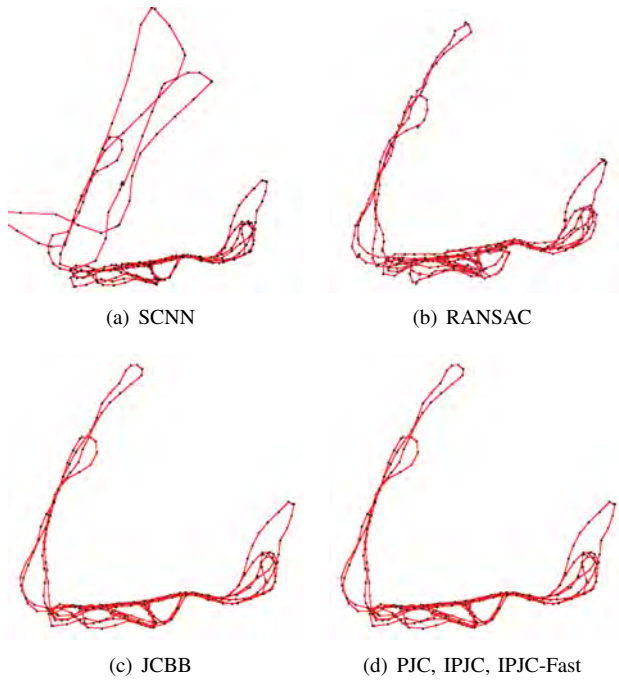


Fig. 7. Victoria Park posterior maps. We used each data association algorithm to produce a pose graph that was then optimized using sparse Cholesky decomposition. The graph generated by SCNN fails to converge due to erroneous data associations. The remaining methods generated visually reasonable graphs (visually indistinguishable in the case of PJC, IPJC, and IPJC-Fast), though a numerical comparison to ground-truth shows that IPJC-Fast's map was the most accurate.

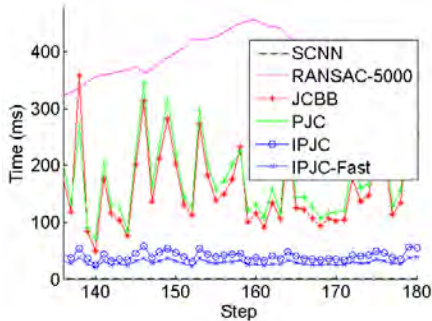


Fig. 8. Victoria Park Computational Complexity. IPJC-Fast was the fastest of the methods that produced a reasonable map. SCNN ran in less time, but the resulting map diverged due to data association errors.

## V. CONCLUSION

We have presented IPJC-Fast, a new method for computing data associations that is both fast and accurate. It is equivalent to the gold-standard JCBB on linear problems, but is formulated in terms of the posterior distribution. It exploits the probabilistic structure present in feature cloud matching, a task common in both SLAM and in object recognition, to achieve significant speed savings over JCBB. Further, by updating the posterior distribution at each level of the interpretation tree, IPJC-Fast computes more accurate compatibility costs.

We demonstrated IPJC-Fast's performance in both simulation and on real data. With the help of ground-truth data,

we were able to show that the accuracy of the compatibility scores were significantly more consistent with those predicted by a  $\chi^2$  distribution. We also demonstrated our method on the Victoria Park dataset, illustrating that it is effective on real-world data.

On feature-cloud matching problems, IPJC-Fast represents significant improvements over existing methods, including JCBB, RANSAC, and SCNN. Reference implementations are available at the authors' website, <http://april.eecs.umich.edu>.

## VI. ACKNOWLEDGMENTS

This work was supported by NSFC grant 61105090 and U.S. DoD Grant FA2386-11-1-4024.

## REFERENCES

- [1] U. Frese, "Closing a million-landmarks loop," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing*, submitted, 2006, pp. 5032–5039.
- [2] J. Neira and J. D. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, December 2001.
- [3] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, August 2002.
- [4] E. Olson, "Recognizing places using spectrally clustered local matches," *Robotics and Autonomous Systems*, 2009.
- [5] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework," *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, December 2004.
- [6] E. Olson, "Robust and efficient robotic mapping," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.
- [7] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goeddel, M. Bulic, J. Crossman, and B. Marinier, "Progress towards multi-robot reconnaissance and the MAGIC 2010 competition," *Journal of Field Robotics*, To appear.
- [8] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Proceedings of the International Symposium of Robotics Research (ISRR)*, O. Faugeras and G. Giralt, Eds., 1988, pp. 467–474.
- [9] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 935–938. [Online]. Available: [citeseer.ist.psu.edu/lu94robot.html](http://citeseer.ist.psu.edu/lu94robot.html)
- [10] P. Maybeck, *Stochastic models, estimation and control*, ser. Mathematics in science and engineering. Academic Press, 1982, no. v. 2.
- [11] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [12] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1066–1077, oct. 2008.
- [13] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "Point feature extraction on 3D range scans taking into account object boundaries," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.
- [14] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.
- [15] J. E. Guivant, F. R. Masson, and E. M. Nebot, "Simultaneous localization and map building using natural features and absolute information," *Robotics and Autonomous Systems*, vol. 40, no. 2-3, pp. 79–90, 2002.
- [16] F. Dellaert, "Square root SAM," in *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, USA, June 2005.

# Inference on Networks of Mixtures for Robust Robot Mapping

Edwin Olson                      Pratik Agarwal  
ebolson@umich.edu    agarwal@informatik.uni-freiburg.de

May 7, 2013

## Abstract

The central challenge in robotic mapping is obtaining reliable data associations (or “loop closures”): state-of-the-art inference algorithms can fail catastrophically if even one erroneous loop closure is incorporated into the map. Consequently, much work has been done to push error rates closer to zero. However, a long-lived or multi-robot system will still encounter errors, leading to system failure.

We propose a fundamentally different approach: allow richer error models that allow the probability of a failure to be explicitly modeled. In other words, rather than characterizing loop closures as being “right” or “wrong”, we propose characterizing the error of those loop closures in a more expressive manner that can account for their non-Gaussian behavior. Our approach leads to an fully-integrated Bayesian framework for dealing with error-prone data. Unlike earlier multiple-hypothesis approaches, our approach avoids exponential memory complexity and is fast enough for real-time performance.

We show that the proposed method not only allows loop closing errors to be automatically identified, but also that in extreme cases, the “front-end” loop-validation systems can be unnecessary. We demonstrate our system both on standard benchmarks and on the real-world datasets that motivated this work.

## 1 Introduction

Robot mapping problems are often formulated as an inference problem on a factor graph: variable nodes (representing the location of robots or other landmarks in the environment) are related through factor nodes, which encode geometric relationships between those nodes. Recent Simultaneous Localization and Mapping (SLAM) algorithms can rapidly find maximum likelihood solutions for maps, exploiting both fundamental improvements in the understanding of the structure of mapping problems [Newman, 1999, Frese, 2005, Dellaert, 2005], and the computational convenience afforded by assuming that error models are simple uni-modal Gaussian [Smith et al., 1988].

Despite their convenience, Gaussian error models often poorly approximate the truth. In the SLAM domain, perceptual aliasing can lead to incorrect loop closures, and the resulting error can lead to divergence of the map estimate. Similarly, the wheels of a robot may sometimes grip and

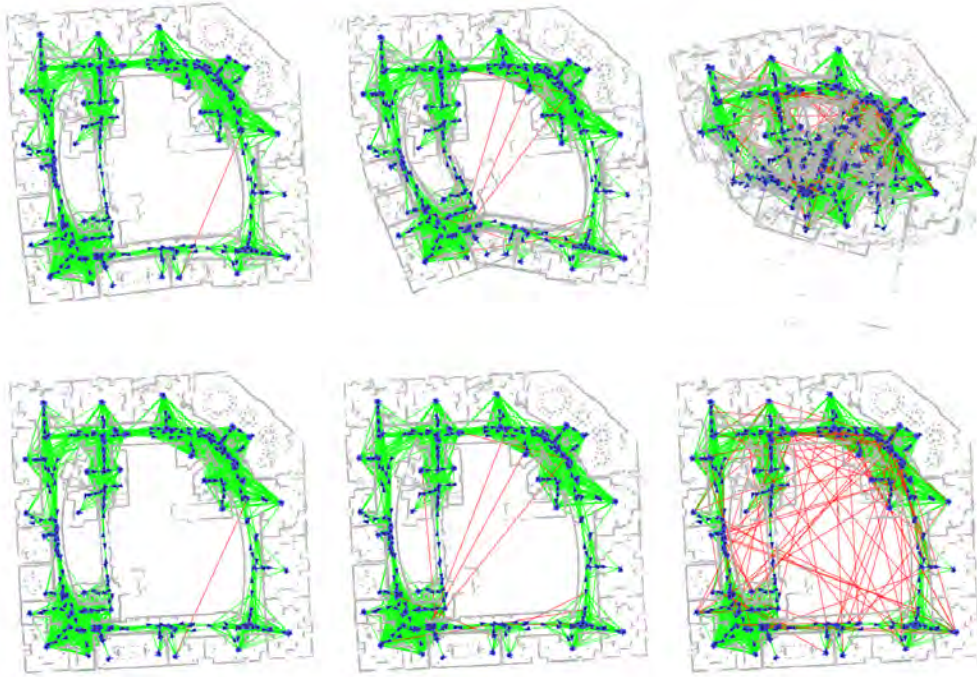


Figure 1: Recovering a map in the presence of erroneous loop closures. We evaluated the robustness of our method by adding erroneous loop closures to the Intel data set. The top row reflects the posterior map as computed by a state-of-the-art sparse Cholesky factorization method with 1, 10, and 100 bad loop closures. The bottom row shows the posterior map for the same data set using our proposed max mixture method. While earlier methods produce maps with increasing global map deformation, our proposed method is essentially unaffected by the presence of the incorrect loop closures.

sometimes slip, leading to a bi-modal motion model. Similar challenges arise throughout robotics, including sonar and radar (with multi-path effects), target-tracking (where multiple disjoint hypotheses may warrant consideration), etc.

In the specific case of SLAM, it has become standard practice to decompose the problem into two halves: a “front-end” and “back-end”. The front-end is responsible for identifying and validating loop closures and constructing a factor graph; the back-end then performs inference (often maximum likelihood) on this factor graph. In most of the literature, it is assumed that the loop closures found by the front-end have noise that can be modeled as Gaussian.

For example, the front-end might assert that the robot is now at the same location that it was ten minutes ago (it has “closed a loop”), with an uncertainty of 1 meter. Suppose, however, that the robot was somewhere else entirely— a full 10 meters away. The back-end’s role is to compute the maximum likelihood map, and an error of ten standard deviations is so profoundly unlikely



that the back-end will almost certainly never recover the correct map: it is compelled to distort the map so as to make the erroneous loop closure more probable (see Fig. 1).

The conventional strategy is to build better front-end systems. Indeed, much effort has been devoted to creating better front-end systems [Neira and Tardos, 2001, Bailey, 2002, Olson, 2009b], and these approaches have succeeded in vastly reducing the rate of errors. But for systems that accumulate many robot-hours of operation, or robots operating in particularly challenging environments, even an extremely low error rate still results in errors. These errors lead to divergence of the map and failure of the system.

Our recent efforts at building a team of robots that can cooperatively explore and map an urban environment [Olson et al., 2012] illustrate the challenges, and motivated this work. At the time, we modeled the uncertainty of odometry and loop closing edges with simple Gaussians, but despite extensive validation of these edges prior to optimization, some of these edges had large errors that were virtually impossible given their noise model. Even with a novel interface allowing a human to help untangle the resulting map [Crossman et al., 2012], errors were still evident (see Fig. 7). Our subsequent analysis revealed that odometry edges were often to blame. We had assumed a 15% noise model, but our robots, driving under autonomous control, would occasionally get caught on small, unsensed obstacles. As a result, the robot actually encountered 100% error—five standard deviations given our prior noise model. The resulting error in our position estimates exacerbated the perceptual aliasing problem: our incorrect position prior would argue against correct loop closure hypotheses, and would favor some incorrect hypotheses.

In this paper, we propose a novel approach that allows efficient maximum-likelihood inference on factor graph networks that contain arbitrarily complex probability distributions. This is in contrast to state-of-the-art factor graph based methods, which are limited to uni-modal Gaussian distributions, and which suffer from the real-world problems described above. Specifically, we propose a new type of mixture model, a *max*-mixture, which provides similar expressivity as a sum-mixture, but avoids the associated computational costs. With such a mixture, the “slip or grip” odometry problem can be modeled as a multi-modal distribution, and loop closures can be accompanied by a “null” hypothesis. In essence, the back-end optimization system serves as a part of the front-end— playing an important role in validating loop closures and preventing divergence of the map.

We will demonstrate our system on real data, showing that it can easily handle the error rates of current front-end data validation systems, allowing robust operation even when these systems

produce poor output. We will also illustrate that, in extreme cases, *no* front-end loop validation is required at all: all candidate loop closures can simply be added to the factor graph, and our approach simultaneously produces a maximum likelihood map while identifying the set of edges that are correct. This is an interesting development, since it provides a fully integrated Bayesian treatment of both mapping and data association, tasks which are usually decoupled.

It has previously been shown that exact inference on even poly-trees of mixtures is NP-hard [Lerner and Parr, 2006]. Our method avoids exponential complexity at the expense of guaranteed convergence to the maximum likelihood solution. In this paper, we explore the robustness of our method, and characterize the error rates that can be handled.

In short, the contributions of this paper are:

- We formulate a new mixture model that provides significant computational advantages over the more traditional sum-of-Gaussians mixtures, while retaining similar expressive power.
- We develop an algorithm for fast maximum-likelihood inference on factor graph networks containing these max-mixtures.
- We demonstrate how robot mapping systems can use these methods to robustly handle errors in odometry and loop-closing systems.
- We characterize the robustness of our method to local minima, identifying factors (like error rate and overall graph degree) and their impact. We show that the basin of convergence is large for a variety of benchmark 2D and 3D datasets over a range of plausible parameter values.
- We evaluate our algorithm on real-world datasets to demonstrate its practical applicability both in terms of the quality of results and the computation time required.

## 2 Related Work

We are not the first to consider estimation in the presence of non-Gaussian noise. Two well-known methods allow more complex error models to be used: particle filter methods and multiple hypothesis tracking (MHT) approaches.

Particle filters, perhaps best exemplified by FastSLAM [Montemerlo, 2003], approximate arbitrary probability distributions through a finite number of samples. Particle filters attempt to

explicitly (and non-parametrically) describe the posterior distribution. Unfortunately, the posterior grows in complexity over time, requiring an ever-increasing number of particles to maintain the quality of the posterior approximation. This growth quickly becomes untenable, forcing practical implementations to employ particle resampling techniques [Hähnel et al., 2003, Kwak et al., 2007, Stachniss et al., 2005]. Unavoidably, resampling leads to a loss of information, since areas with low probability density are effectively truncated to zero. This loss of information can make it difficult to recover the correct solution, particularly after a protracted period of high uncertainty [Bailey et al., 2006, Grisetti et al., 2005].

Multiple Hypothesis Tracking approaches [Durrant-Whyte et al., 2003, Blackman, 2004] provide an alternative approach more closely related to mixture models. These explicitly represent the posterior using an ensemble of Gaussians that collectively encode a mixture. However, the size of the ensemble also grows rapidly: the posterior distribution arising from  $N$  observations each with  $c$  components is a mixture with  $c^N$  components. As with particle filters, this exponential blow-up quickly becomes intractable, forcing approximations that cause information loss and ultimately lead to errors.

In the special case where errors are modeled as uni-modal Gaussians, the maximum likelihood solution of the factor graph network can be found using non-linear least squares. Beginning with the observation that the information matrix is sparse [Thrun and Liu, 2003, Walter et al., 2005, Eustice et al., 2006], efforts to exploit that sparsity resulted in rapid improvements to map inference by leveraging sparse factorization and good variable-ordering heuristics [Dellaert and Kaess, 2006, Kaess et al., 2008, Konolige, 2010, Agarwal and Olson, 2012]. While the fastest of these methods generally provide only maximum-likelihood inference (a shortcoming shared by our proposed method), approximate marginal estimation methods are fast and easy to implement [Bosse et al., 2004, Olson, 2008]. It is highly desirable for new methods to be able to leverage the same insights that made these approaches so effective.

Sum-mixtures of Gaussians have been recently been explored [Pfingsthorn and Birk, 2012]. The mixtures are converted into uni-modal Gaussians via a “pre-filtering” step, yielding a problem that can be approximately solved using standard sparse methods. Another approach for increasing robustness is to use the  $\chi^2$  of individual measurements in order to identify clusters of mutually-consistent loop closures [Latif et al., 2012b]. This mutual consistency can be re-evaluated as new information arrives [Latif et al., 2012a]. The “max-mixture” approach described in this paper differs from these approaches in that the challenging process of approximating a sum-mixture is

avoided, and that the set of activated modes is intrinsically re-evaluated at every iteration.

One method similar to our own explicitly introduces switching variables whose value determines whether or not a loop closure is accepted [Sunderhauf and Protzel, 2012]. This work is notable for being the first to propose a practical way of dealing with front-end errors. In comparison to our approach, they penalize the activation/deactivation of a loop closure through a separate cost function (as opposed to being integrated into a mixture model), and must assign initial values to these switching variables (as opposed to our implicit inference over the latent variables). Our approach does not introduce switching variables, instead explaining poor quality data in the form of a non-Gaussian probability density function which can be arbitrarily complex (including having multiple maxima).

Robustified cost functions [Hartley and Zisserman, 2004] provide resilience to errors by reducing the cost associated with outliers, and have been widely used in the vision community [Strasdat et al., 2010, Sibley et al., 2009]. Our proposed max-mixture model can approximate arbitrary probability distributions, including those arising from robustified cost functions.

Our proposed method avoids the exponential growth in memory requirements of particle filter and MHT approaches by avoiding an explicit representation of the posterior density. Instead, like other methods based on sparse factorization, our method extracts a maximum likelihood estimate. Critically, while the memory cost of representing the posterior distribution grows exponentially, the cost of storing the underlying factor graph network (which implicitly encodes the posterior) grows only linearly with the size of the network. In other words, our method (which only stores the factor graph) can recover solutions that would have been culled by particle and MHT approaches. In addition, our approach benefits from the same sparsity and variable-ordering insights that have recently benefited uni-modal approaches.

### 3 Approach

Our goal is to infer the posterior distribution of the state variable (or map)  $x$ , which can be written in terms of the factor potentials in the factor graph. The probability is conditioned on sensor observations  $z$ ; with an application of Bayes' rule and by assuming an uninformative prior  $p(x)$ , we obtain:

$$p(x|z) \propto \prod_i p(z_i|x) \tag{1}$$

Prior to this work, it is generally assumed that the factor potentials  $p(z_i|x)$  can be written as

Gaussians:

$$p(z_i|x) = \frac{1}{(2\pi)^{n/2}|\Lambda_i^{-1}|^{1/2}} e^{-\frac{1}{2}(f_i(x)-z_i)^T \Lambda_i (f_i(x)-z_i)} \quad (2)$$

Further, while  $f_i(x)$  is generally non-linear, it is assumed that it can be approximated using a first-order Taylor expansion such that  $f_i(x) \approx J_i \Delta x + f_i(x_0)$ .

The posterior maximum likelihood value can be easily solved in such cases by taking the logarithm of Eqn. 1, differentiating with respect to  $x$ , then solving for  $x$ . This classic least-squares approach leads to a simple linear system of the form  $Ax = b$ . Critically, this is possible because the logarithm operator can be “pushed” inside the product in Eqn. 1, reducing the product of  $N$  terms into a sum of  $N$  simple quadratic terms. No logarithms or exponentials remain, making the resulting expression easy to solve.

We might now consider a more complicated function  $p_i(x|z)$ , such as a sum-mixture of Gaussians:

$$p(z_i|x) = \sum_i w_i N(\mu_i, \Lambda_i^{-1}) \quad (3)$$

In this case, each  $N(\mu_i, \Lambda_i^{-1})$  represents a different Gaussian distribution. Such a sum-mixture allows great flexibility in specifying the distribution  $p(z_i|x)$ . For example, we can encode a robustified cost function by using two components with the same mean, but with different variances. More complicated distributions, including those with multiple maxima, can also be represented.

The problem with a sum-mixture is that the maximum likelihood solution is no longer simple: the logarithm can no longer be pushed all the way into the individual Gaussian components: the summation in Eqn. 3 prevents it. As a result, the introduction of a sum-mixture means that it is no longer possible to derive a simple solution for  $x$ .

### 3.1 Max-Mixtures

Our solution to this problem is a new mixture model type, one based on a max operator rather than a sum:

$$p(z_i|x) = \max_i w_i N(\mu_i, \Lambda_i^{-1}) \quad (4)$$

While the change is relatively minor, the implications to optimization are profound. The logarithm *can* be pushed inside a max mixture: the max operator acts as a selector, returning a single well-behaved Gaussian component.

A max mixture has much of the same character as a sum mixture and retains a similar ex-

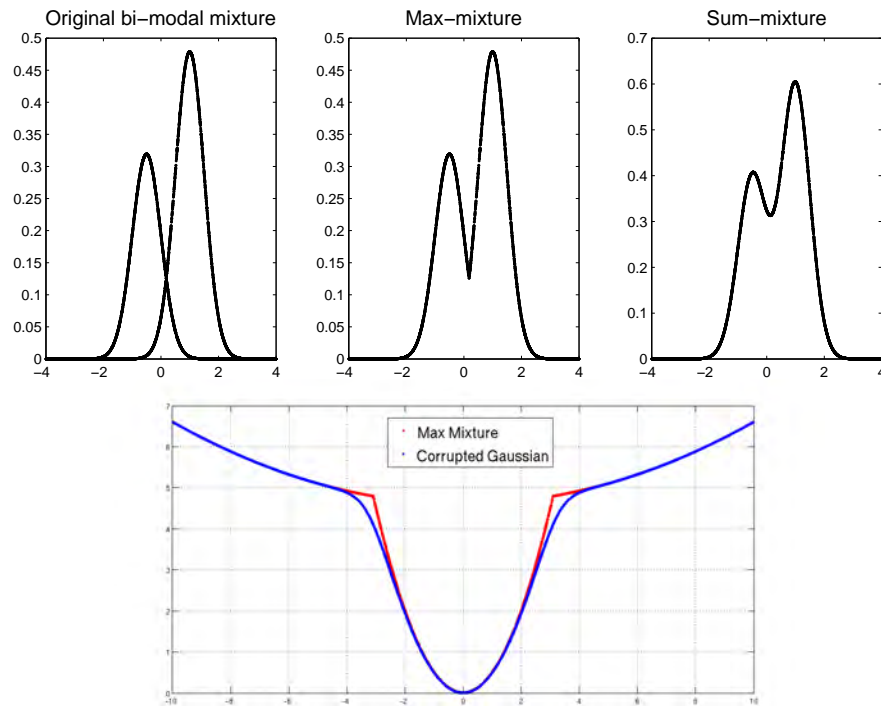


Figure 2: Mixture Overview. Given two mixture components (top left), the max- and sum- mixtures produce different distributions. In both cases, arbitrary distributions can be approximated. A robustified cost function (in this case a corrupted Gaussian, bottom) can be constructed from two Gaussian components with equal means but different variances.

pressivity: multi-modal distributions and robustified distributions can be similarly represented (see Fig. 2). Note, however, that when fitting a mixture to a desired probability distribution, different components will result for sum- and max- mixtures. Assuring that the distributions integrate to one is also handled differently: for a sum mixture,  $\sum w_i = 1$  is a necessary and sufficient condition; for a max mixture, proper normalization is generally more difficult to guarantee. Usefully, for maximum likelihood inference, it is inconsequential whether the distribution integrates to 1. Specifically, suppose that some normalization factor  $\gamma$  is required in order to ensure that a max mixture integrates to one. Since  $\gamma$  is used to scale the distribution, the log of the resulting max mixture is simply the log of the un-normalized distribution plus a constant. The addition of such a constant does not change the solution of a maximum-likelihood problem, and thus it is unnecessary for our purposes to compute  $\gamma$ .

### 3.2 Cholesky-MM

We now show how max mixture distributions can be incorporated into existing graph optimization frameworks. The principle step in such a framework is to compute the Jacobian, residual, and information matrix for each factor potential. As we described previously, these are trivial to compute for a uni-modal Gaussian distribution.

For the max-mixture case, it might seem that computing the needed derivatives for the Jacobian is difficult: the max-mixture is not actually differentiable at the points where the maximum-likelihood component changes. While this makes it difficult to write a closed-form expression for the derivatives, they are none-the-less easy to compute.

The key observation is that the max operator serves as a selector: once the mixture component with the maximum likelihood is known, the behavior of the other mixture components is irrelevant. In other words, the solver simply iterates over each of the components, identifies the most probable, then returns the Jacobian, residual, and information matrix for that component. If the likelihood of two components are tied—an event which corresponds to evaluating the Jacobian at a non-differentiable point—we pick one of the components arbitrarily. However, these boundary regions comprise an area with zero probability mass.

The resulting Jacobians, residuals, and information matrices are combined into a large least-squares problem which we subsequently solve with a minimum-degree variable ordering heuristic followed by sparse Cholesky factorization using Gauss-Newton steps, in a manner similar to that described by [Dellaert, 2005]. With our modifications to handle max-mixtures, we call our system

Cholesky-MM.

It is often necessary to iterate the full least-squares problem several times. Each time, the best component in each max-mixture is re-evaluated: in essence, as the optimization proceeds, we dynamically select the best mixture component as an integral part of the optimization process.

Even in the non-mixture case, this sort of non-linear optimization cannot guarantee convergence to the global optimal solution. It is useful to think of a given inference problem as having a “basin of convergence”—a region that contains all the initial values of  $x$  that would ultimately converge to the global optimal solution. For most well-behaved problems with simple Gaussian distributions, the basin of convergence is large. Divergence occurs when the linearization error is so severe that the gradient points in the wrong direction.

The posterior distribution for a network with  $N$  mixtures, each with  $c$  components, is a mixture with as many as  $c^N$  components. In the worst-case, these could be non-overlapping, resulting in  $c^N$  local minima. The global optimal solution still has a basin of convergence: if our initial solution is “close” to the optimal solution, our algorithm will converge. But if the basin of convergence is extremely small, then the practical utility of our algorithm will be limited.

In other words, the key question to be answered about our approach is whether the basin of convergence is usefully large. Naturally, the size of this basin is strongly affected by the properties of the problem and the robustness of the algorithm used to search for a solution. One of the main results of this paper is to show that our approach yields a large basin of convergence for a wide range of useful robotics problems.

## 4 Applications and Evaluation

In this section, we show how our approach can be applied to several real-world problems. We include quantitative evaluations of the performance of our algorithm, as well as characterize its robustness and runtime performance.

### 4.1 Uncertain loop closures

We first consider the case where we have a front-end that produces loop closures with a relatively low, but non-zero, error rate. For each uncertain loop closure, we introduce a max-mixture consisting of two components: 1) the front-end’s loop closure and 2) a null hypothesis. The null hypothesis, representing the case when the loop closure is wrong, is implemented using a mixture



component with a very large covariance. In our experiments, we set the mean of the null-hypothesis component equal to that of the other component. Weights and variances are assigned to these two components in accordance with the error rate of the front-end.

In practice, the behavior of the algorithm is not particularly sensitive to the weights associated with the null hypotheses. The main benefit of our approach arises from having a larger probability associated with incorrect loop closures, as opposed to the exponentially-fast decreasing probability specified by the loop closer’s Gaussian. Even if the null hypothesis has a very low weight (for example  $10^{-5}$ ), it will provide a sufficiently plausible explanation of the data to prevent a radical distortion of the graph. Second, once the null hypothesis becomes dominant, its large variance results in a very weak gradient for the edge. As a result, the edge plays virtually no role in subsequent optimization. We set the mean of the null hypothesis equal to that of the front-end’s hypothesis so that the small amount of gradient that remains produces a slight bias back towards the front-end’s hypothesis. If subsequent observations re-affirm the front-end’s hypothesis, it can still become active in the future. Unlike particle filter or MHT methods which must eventually truncate unlikely events, no information is lost.

A two-component mixture model in which both components have identical means but different variances can be viewed as a robustified cost function. In particular, parameters can be chosen so that a two-component max mixture closely approximates a corrupted Gaussian [Hartley and Zisserman, 2004] (see Fig. 2).

To evaluate the performance of our approach, we added randomly-generated loop closures to two standard benchmark datasets: the 3500 node Manhattan set [Olson, 2008] and the Intel data set [Howard and Roy, 2003]. These were processed in an online fashion, adding one pose at a time and potentially one or more loop closures (both correct and incorrect). This mimics real-world operation better than a batch approach, and is more challenging due to the fact that it is easier to become caught in a local minimum since fewer edges are available to guide the optimization towards the global optimum.

For a given number of randomly-generated edges, we compute the posterior map generated by our method and a standard non-mixture method, using a laser-odometry solution as the initial state estimate. The mean-squared error of this map is compared to the global optimal solution [Olson, 2011], and listed in Fig. 3.

Our proposed method achieves dramatically lower mean squared errors (MSE)<sup>1</sup> than standard

---

<sup>1</sup>We report MSE based on translational error, i.e.  $\text{MSE}_{xy}$  for 3dof and  $\text{MSE}_{xyz}$  for 6dof problems.

Manhattan Data Set						
True Edges	False Edges	True Pos.	False Pos.	Avg. FP Err.	MSE (Our method)	MSE (Non-mixture)
2099	0	2099	0	NaN	0.6726	0.6726
2099	10	2099	0	NaN	0.6713	525.27
2099	100	2099	1	0.0208	0.6850	839.39
2099	200	2099	2	0.5001	0.6861	874.67
2099	500	2099	3	0.6477	0.6997	888.82
2099	1000	2099	10	0.7155	0.7195	893.98
2099	2000	2099	22	0.5947	0.7151	892.54
2099	3000	2099	36	0.5821	0.7316	896.01
2099	4000	2099	51	0.6155	0.8317	896.05

Intel Data Set						
True Edges	False Edges	True Pos.	False Pos.	Avg. FP Err.	MSE (Our method)	MSE (Non-mixture)
14731	0	14731	0	NaN	$7.122 \times 10^{-10}$	$1.55 \times 10^{-9}$
14731	10	14731	0	NaN	$7.123 \times 10^{-10}$	0.044
14731	100	14731	2	0.1769	$4.431 \times 10^{-6}$	2.919
14731	200	14731	9	0.1960	$5.583 \times 10^{-6}$	8.810
14731	500	14731	19	0.1676	$1.256 \times 10^{-5}$	34.49
14731	1000	14731	29	0.1851	$5.840 \times 10^{-5}$	71.86
14731	2000	14731	64	0.1937	$2.543 \times 10^{-4}$	86.37
14731	3000	14731	103	0.1896	$3.307 \times 10^{-4}$	91.04
14731	4000	14217	146	0.1699	0.014	95.36

Figure 3: Null-hypothesis robustness. We evaluate the robustness of our method and a standard Gaussian method to the presence of randomly-generated edges. As the number of randomly-generated edges increases, the mean squared error (MSE) of standard approaches rapidly degenerates; our proposed method produces good maps even when the number of randomly-generated edges is large in comparison to the number of true edges. Our approach does accept some randomly-generated edges (labeled “false positives” above), however the error of these accepted edges is comparable to that of the true positives. In each case, the initial state estimate is that from the open-loop odometry.

non-mixture versions. While the true positive rate is nearly perfect in both experiments, some randomly-generated edges (labeled false positives) *are* accepted by our system. However, since the false positives are randomly generated, some of them (by chance) are actually close to the truth. Such “accidentally correct” edges *should* be accepted by our algorithm<sup>2</sup>.

We can evaluate the quality of the accepted edges by comparing the error distribution of the true positives and false positives (see Fig. 4). As the histogram indicates, the error distribution is similar, though the error distribution for the false positives is slightly worse than for the true positives. Still, no extreme outliers (the type that cause divergence of the map) are accepted by

<sup>2</sup>We favor generating “false positives” in a purely random way, even though it leads to “accidentally” correct edges. Any filtering operation to reject these low-error edges would introduce a free parameter (the error threshold) whose value could be tuned to favor the algorithm.

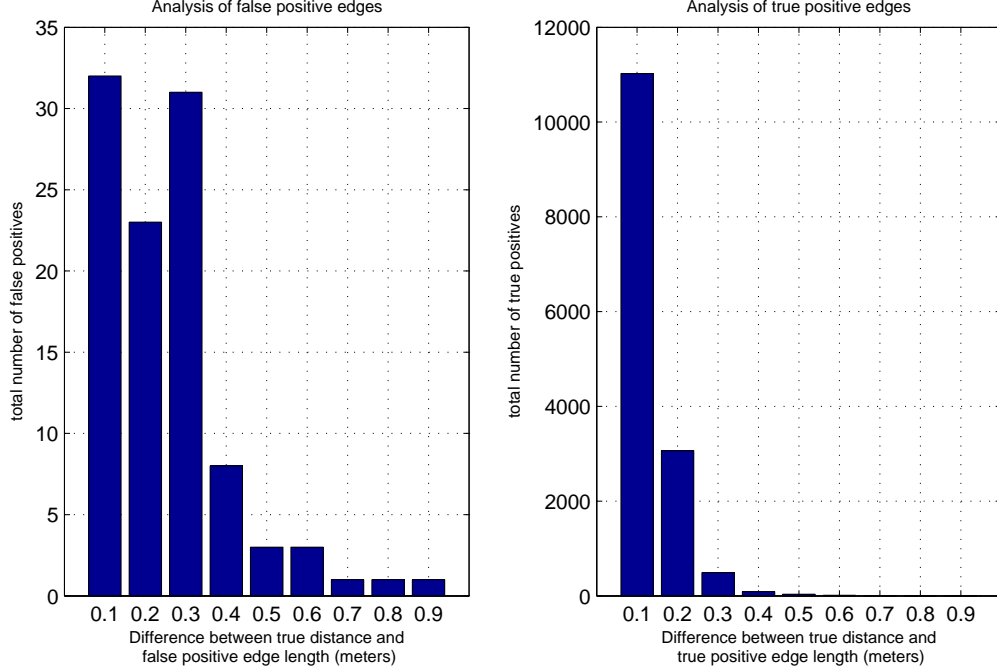


Figure 4: Error distribution for true and false positives. Our method accepts some randomly-generated “false positives”, but an analysis of the error of those edges indicates that they (left) are only slightly worse than the error of true edges (right).

our method.

#### 4.1.1 Extension to 6DOF

While many important domains can be described in terms of planar motion (with three-dimensional factor potentials reflecting translation in x, translation in y, and rotation), there is increasing interest in 6 degree-of-freedom problems. Rotation is a major source of non-linearity in SLAM problems, and full six degree-of-freedom problems can be particularly challenging.

To evaluate the performance of our method on a six degree-of-freedom problem, we used the benchmark Sphere2500 dataset [Kaess et al., 2008]. This dataset does not contain incorrect loop closures, and so we added additional erroneous loop closures. In Fig. 5, we show the results of a standard Cholesky solver and our max mixture approach applied to corrupted Sphere2500 dataset with an additional 1, 10, and 100 erroneous edges. As in previous examples, the maps produced by a standard method quickly deteriorate. In contrast, the proposed method produces posterior maps that are essentially unaffected by the errors. In this experiment, each loop closure edge in the graph (both correct and false) was modeled as a two-component max mixture in which the second component had a large variance ( $10^7$  times larger than the hypothesis itself) and a small

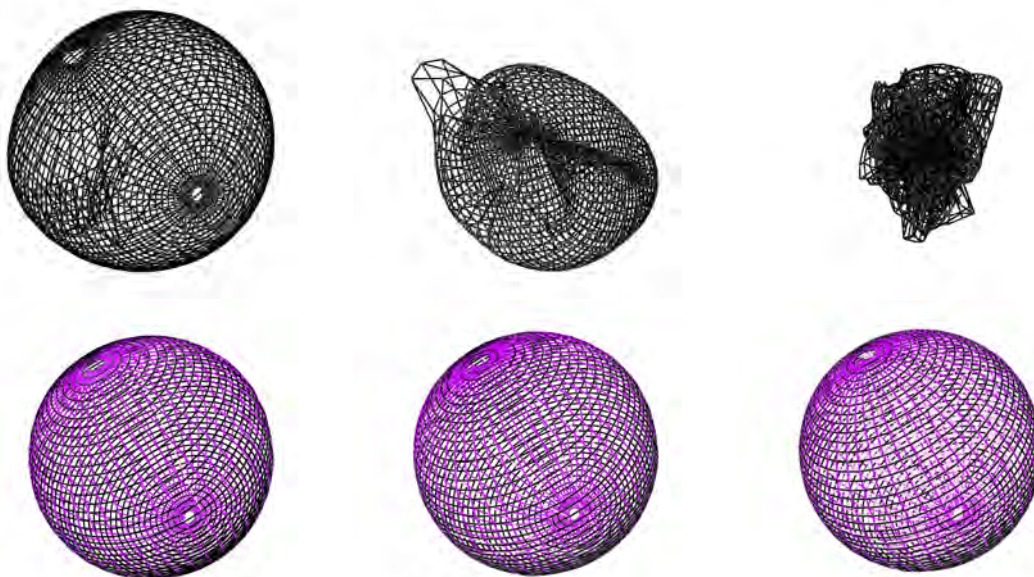


Figure 5: Recovering a map in the presence of outliers. We evaluated the robustness of our method by adding erroneous loop closure edges to the Sphere2500 dataset, a dataset with a full 6 degrees-of-freedom. The top row reflects the posterior of the map with a standard least square Cholesky solver with 1, 10, and 100 wrong edges. The bottom row shows the corresponding map for the same dataset using max mixtures method.

weight ( $10^{-7}$ ). The method is relatively insensitive to the particular values used: the critical factor is ensuring that, if the hypothesis is incorrect, the null hypothesis provides a higher probability explanation than the putative (incorrect) hypothesis, and that the null hypothesis is sufficiently weak so as to not distort the final solution. The impact of the relative strength of the null hypothesis on the basin of convergence is explored experimentally in Sec. 4.5.

## 4.2 Multi-modal distributions

In the previous sections, we demonstrated that our method can be used to encode null hypotheses, or equivalently, implement robustified cost functions—a capability similar to earlier work [Sunderhauf and Protzel, 2005]. In that case, the probability distributions being approximated by each mixture have only a single maximum. Our use of a mixture model, however, also allows multi-modal distributions to be encoded. The ability to directly represent multi-modal distributions is a feature of our approach.

### 4.2.1 Slip or Grip problem

One of the original motivating problems of this work was dealing with the “slip or grip” problem: the case where a robot’s wheels occasionally slip catastrophically, resulting in near zero motion. With a typical odometry noise model of 10-20%, such an outcome would wreak havoc on the posterior map.

Our approach to the “slip or grip” problem is to use a two-component mixture model: one component (with a large weight) corresponds to the usual 15% noise model, while the second component (with a low weight) is centered around zero. No changes to our optimization algorithm are required to handle such a case. However, since the distribution now has multiple local maxima, it poses a greater challenge in terms of robustness.

Of course, without some independent source of information that contradicts the odometry data, there is no way to determine that the wheels were slipping. To provide this independent information, we used a state-of-the-art scan matching system [Olson, 2009a] to generate loop closures. We manually induced wheel slippage by pulling on the robot. Despite the good loop closures, standard methods are unable to recover the correct map. In contrast, our method determines that “slip” mode is more likely than the “grip” mode, and recovers the correct map (see Fig. 6).

As part of our earlier multi-robot mapping work [Ranganathan et al., 2010, Olson et al., 2012],

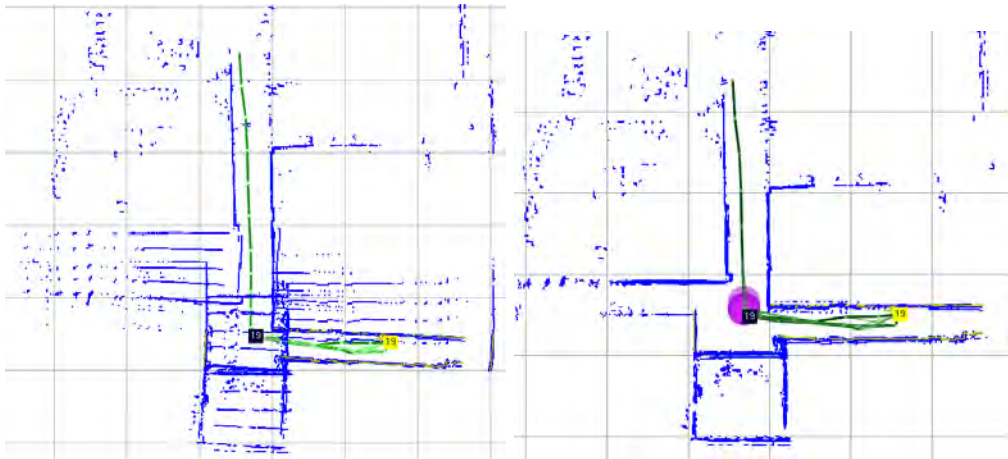


Figure 6: Slip or Grip Example. We evaluate the ability of our algorithm to recover a good map in the presence of catastrophically slipping wheels. In this case, the robot is obtaining loop closures using a conventional laser scanner front-end. These loop closures are of high quality, but the odometry edges still cause significant map distortions when using standard methods (left). When a small probability is added to account for slippage, our mixture approach recovers a much improved map (right).

we employed a team of 14 robots to explore a large urban environment. Wheel slippage contributed to a poor map in two ways: 1) the erroneous factor potentials themselves, and 2) the inability to identify good loop closures due to a low quality motion estimate. By using a better odometry model, our online system produced a significantly improved map (see Fig. 7).

#### 4.2.2 Simplifying the Front End with “one-of-k” Formulation

In current approaches, front-end systems are typically responsible for validating loop closures prior to adding them to the factor graph network. However, if the back-end can recover from errors, is it possible to omit the filtering entirely?

In certain cases, our inference method can eliminate the need for loop validation by the front-end. This is desirable from a conceptual standpoint: in principle, a better map should result from handling loop closing and mapping from within an integrated Bayesian framework. The conventional decoupling of mapping into a front-end and back-end, while practical, prevents a fully Bayesian solution.

We evaluated this possibility using the Intel data set. At every pose, a laser scan matcher attempts a match to *every* previous pose. The top  $k$  matches (as measured by overlap of the two scans) are formed into a mixture containing  $k + 1$  components. (The extra component remains a null hypothesis to handle the case where all  $k$  matches are incorrect.) To push the experiment as far as possible, no position information was used to prune the set of  $k$  matches. Larger values of

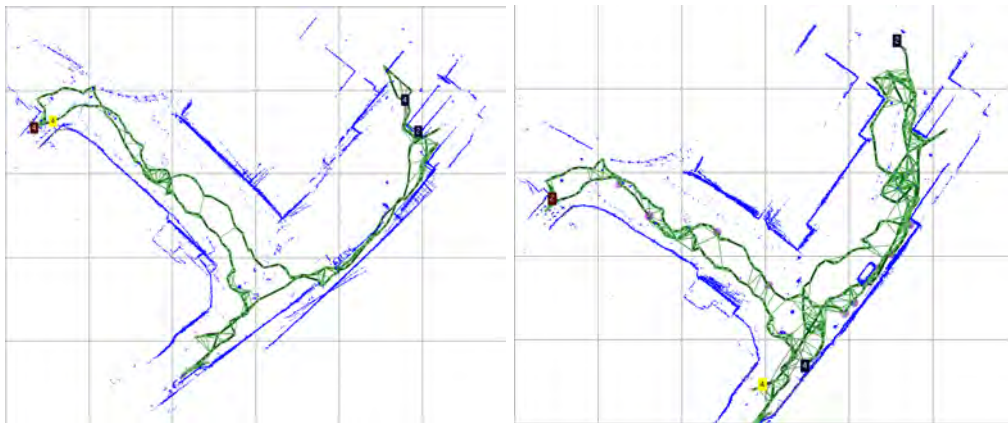


Figure 7: Online results using odometry mixture model. The left figure shows a map of a  $30\text{m} \times 25\text{m}$  area in which our multi-robot urban mapping team produced a poor map due to wheel slippage and the ensuing inability to find loop-closures. With our odometry mixture model (right), the wheel slippage is (implicitly) detected, and we find additional loop closures. The result is a significantly improved map.

$k$  provide robustness against perceptual aliasing, since it increases the likelihood that the correct match is present somewhere within the set of  $k$  components.

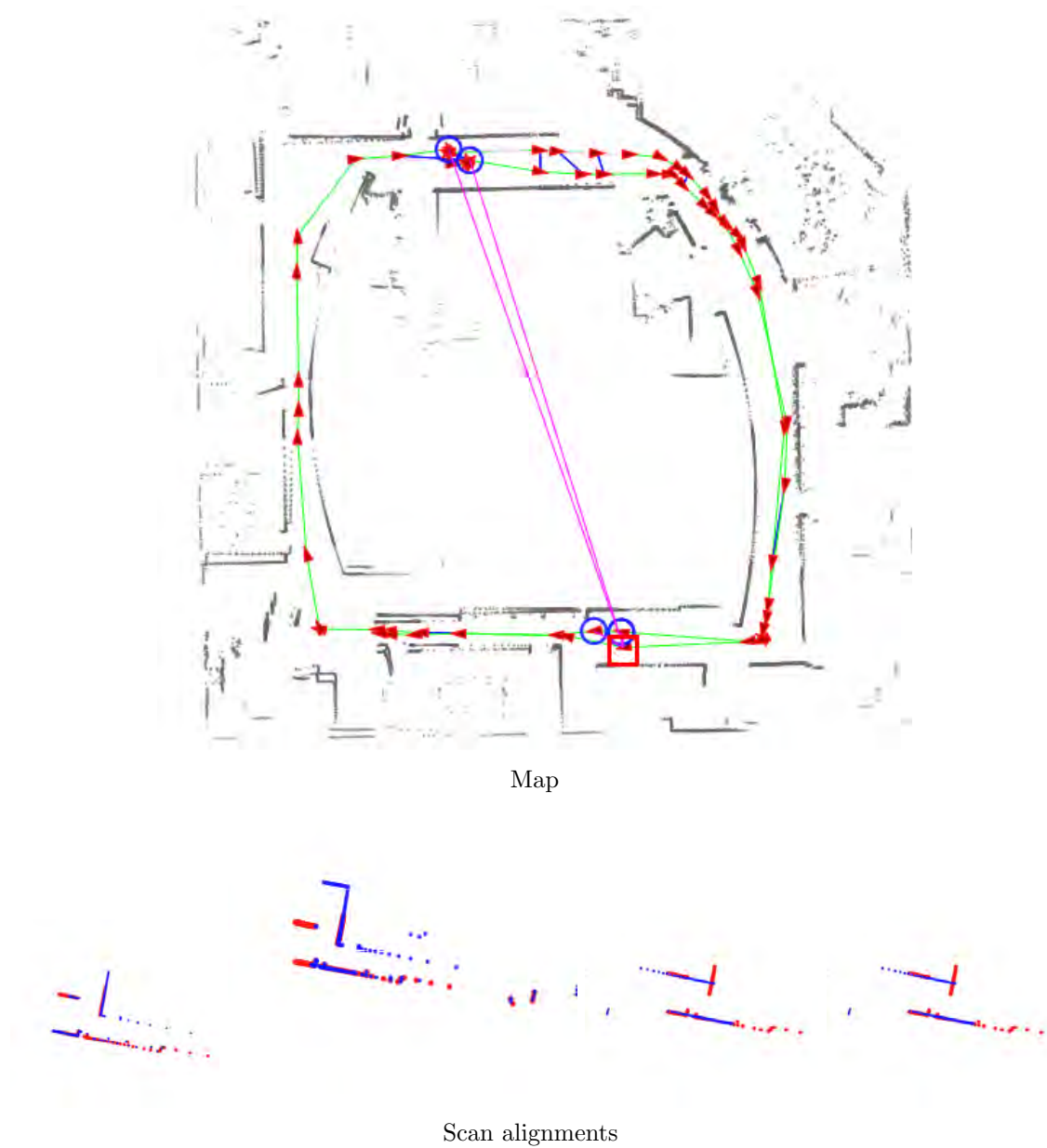


Figure 8: Data-association as a mixture. Given a query pose (red square at bottom of map), we perform a brute-force scan matching operation to all previous poses. The best 4 scan match results, based on overlap, are added to a max-mixture model that also includes a null hypothesis. The position of the best matches are shown as blue circles, and the corresponding scan matches shown at the bottom. The similarity in appearance between the blue poses represents a significant degree of perceptual aliasing. The scan matcher finds two correct matches and two incorrect matches. The two correct matches are the two blue circles at the bottom of the map and the first two scan alignments.



An example of one mixture with  $k = 4$  putative matches is shown in Fig. 8. The weight of the components is set in proportion to the score of the scan matcher.

Running our system in an online fashion, we obtain the final map shown in Fig. 9. Online operation is more difficult than batch operation, since there is less information available early on to correct erroneous edges. Our system recovers a consistent global map despite the lack of any front-end loop validation.

The quality of the open-loop trajectory estimate plays an important role in determining whether the initial state estimate is within the basin of convergence. In this case, our open-loop trajectory estimate is fairly good, and our method is able to infer the correct mode for each mixture despite the lack of any front-end loop validation.

The robustness of our method is amplified by better front-end systems: with better quality loop closures, the basin of convergence is enlarged, allowing good maps to be computed even when the open-loop trajectory is poor.

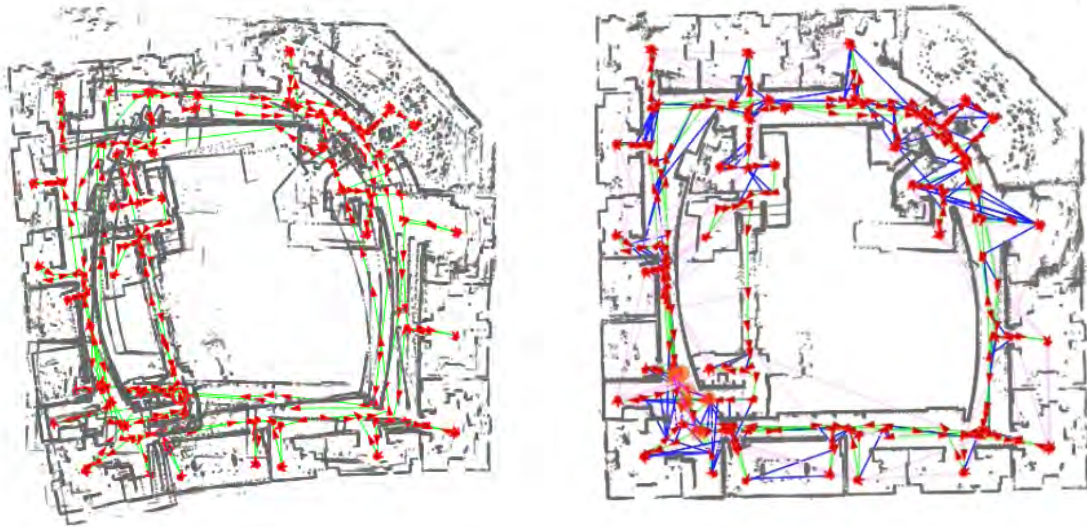


Figure 9: Intel without front-end loop validation. Our system can identify correct loop closures and compute a posterior map from within a single integrated Bayesian framework (right); the typical front-end loop validation has been replaced with a  $k + 1$  mixture component containing the  $k$  best laser scan matches (based purely on overlap) plus a null hypothesis. In this experiment, we used  $k = 5$ . For reference, the open-loop trajectory of the robot is given on the left.



### 4.2.3 Performance Impact of Uncertainty Modeling

In the previous section, uncertain data associations were modeled as “one-of- $k$ ” mixtures, in which multiple candidate loop closures were grouped together in a single edge. Alternatively, each candidate loop closure could be encoded as a two-component mixture in a “null-hypothesis” style mixture; this approach is well-suited to the case where little is known about alternatives to a putative loop closure, while still allowing for the possibility that it is incorrect. (It is also possible that the mixture components have no obvious semantic meaning: the mixture model could simply be approximating a more complex distribution. For example, a max mixture could be fit to an empirically derived cost function from a correlation-based scan matcher [Olson, 2009a]).

In this section, we explore the performance impact of “one-of- $k$ ” mixtures versus “null-hypothesis” mixtures. Consider a “one-of- $k$ ” mixture consisting of three candidate loop closures plus a null hypothesis:  $\{L_1, L_2, L_3, \text{null}\}$ . This can be transformed into three “null-hypothesis” mixtures:  $\{L_1, \text{null}\}$ ,  $\{L_2, \text{null}\}$ , and  $\{L_3, \text{null}\}$ . These two formulations are not exactly equivalent: the “one-of- $k$ ” encodes mutual-exclusion between the hypotheses, whereas the  $k$  separate “null-hypotheses” would permit solutions in which more than one of the loop closures was accepted. In many practical situations, however, the semantic difference is relatively minor. In this section, we show that the performance impact of this choice can be dramatic.

In Table 1, we show results from an experiment in which both formulations were used. We consider the case where loop hypotheses are generated in pairs and in triples; this leads to “one-of- $k$ ” mixtures with three and four components respectively once a null hypothesis is added. For the “one-of- $k$ ” formulation, the null hypotheses has a mean chosen randomly from one of the  $k$  constraints and a large variance roughly the size of the whole map. An alternative graph, constructed from “null-hypothesis” mixtures is constructed from the same sets of loop closure hypotheses; naturally, each of these has two components.

An obvious difference between the two formulations is the number of edges in the graph: the “null-hypothesis” approach creates many more edges. That alone can be expected to increase computational time versus a “one-of- $k$ ” encoding. However, a more critical scaling issue becomes apparent: the “null-hypothesis” formulation leads to dramatically higher fill-in due to the fact that more nodes are connected to factor potentials. In contrast, a “one-of- $k$ ” edge does not contribute the same fill-in, since only one of the components in the mixture has any effect during a single Cholesky iteration. In other words, the max operator in the max mixture formulation effectively

Dataset		Switchable constraints	bi-modal MM	k-modal MM
manhattan with $k = 2$ outliers = 2099	iter time (s)	0.90 s	0.74 s	0.13 s
	fill-in (%)	1.50 %	2.89 %	0.17 %
	#loop edges	4198	4198	2099
	#components	-	2	3
manhattan with $k = 3$ outliers= 4198	iter time (s)	1.5 s	1.2 s	0.13 s
	fill-in (%)	1.70 %	4.30 %	0.17 %
	#loop edges	6277	6277	2099
	#components	-	2	4

Table 1: Runtime comparison between switchable constraints, “null-hypothesis”, “one-of-k” formulations. Groups of related hypotheses were generated and either grouped as a single set of mutually-exclusive edges (one-of-k), individually associated with a null hypothesis, or individually associated with a switching variable [Sunderhauf and Protzel, 2012]. Using the one-of-k formulation reduces the effective connectivity in the graph, reducing fill-in, and resulting in faster computation time.

severs edges corresponding to sub-dominant mixture components, improving the sparsity of the information matrix.

The difference in fill-in leads to significant increases in runtime: on the Manhattan-3500 dataset with groups of three candidate hypotheses, moving from a “one-of-k” to a “null-hypothesis” formulation causes an increase in non-zero entries from 0.17% to 4.3%, with a corresponding increase in computation time from 0.13 s to 1.2 s.

Table 1 also reports runtimes for Sünderhauf’s switchable constraints approach [Sunderhauf and Protzel, 2011] which adds an additional “switching” variable for every edge. In this way, it is semantically comparable to the “null-hypothesis” approach, though the formulation is somewhat different. The runtime of the switchable constraints approach, 1.5 s, is somewhat worse than “null-hypothesis” approach and much worse than the “one-of-k” approach. (Note, for this comparison, all methods were implemented in the g2o [Kummerle et al., 2011] framework using *CHOLMOD* with a *COLAMD* variable ordering.)

These results suggest that, when semantically reasonable to do so, it is preferable to use “one-of-k” mixtures rather than either “null-hypothesis” mixtures or switchable constraints.

### 4.3 Robustness

We have identified two basic factors that have a significant influence on the success of our method: the number of incorrect loop closures and the node degree of the graph. The node degree is an important factor because it determines how over-determined the system is: it determines the degree

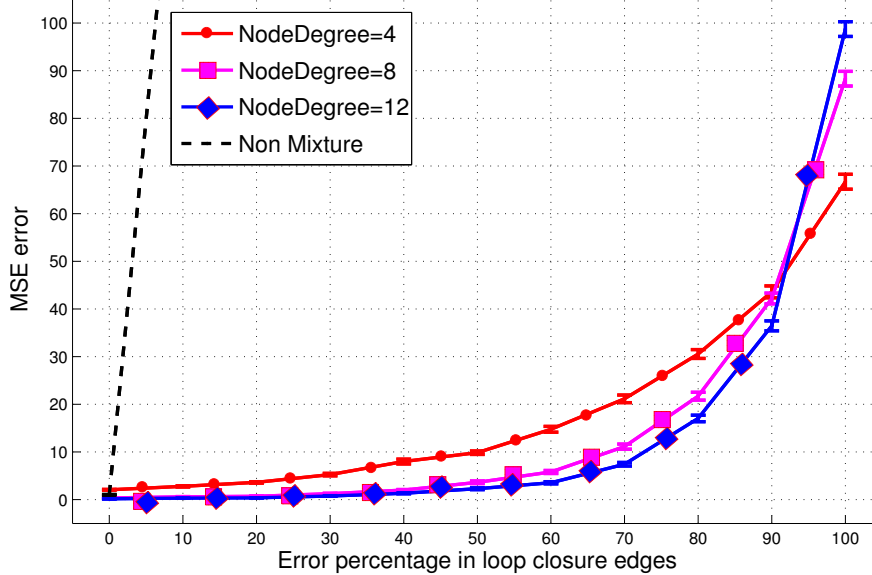


Figure 10: Effect of error rate and node degree on robustness. We evaluate the quality of posterior maps (in terms of mean squared error) as a function of the percentage of bad edges and the node degree of the graph. Each data point represents the average of 3,000 random trials; the standard error is also plotted showing that the results are significant. The quality of the posterior graph is robust to even high levels of error, and is improved further by problems with a high node degree. Our methods, regardless of settings, dramatically out-perform non-mixture methods (black dotted line).

to which correct edges can “overpower” incorrect edges.

To illustrate the relationship between these factors and the resulting quality of the map (measured in terms of mean squared error), we considered a range of loop-closing error rates (ranging from 0% to 100%) for graphs with an average node degree of 4, 8, and 12. Note that an error rate of 80% means that incorrect loop closures outnumber correct loop closures by a ratio of 4:1. In each case, the vehicle’s noisy odometry is also provided. For each condition, we evaluate the performance of our method on 100,000 randomly-generated Manhattan-world graphs (see Fig. 10). Our method produces good maps even when the error rate is very high, and the performance improves further with increasing node degree. In contrast, a standard non-mixture approach diverges almost immediately.

#### 4.4 Runtime Performance

The performance of our method is comparable to existing state-of-the-art sparse factorization methods (see Fig. 11). It takes additional time to identify the maximum likelihood mode for each mixture, but this cost is minor in comparison to the cost of solving the resulting linear system.

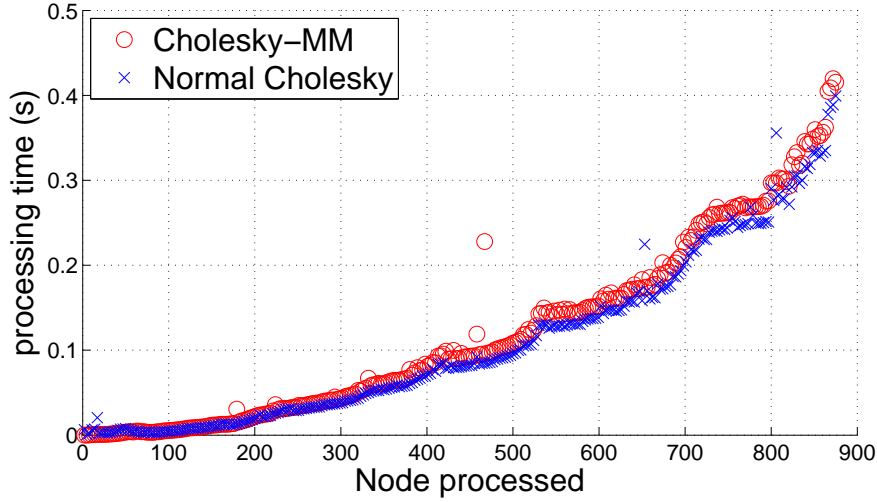


Figure 11: Runtime performance. Using the Intel dataset, we plot the time required to compute a posterior map after every pose, using a batch solver. Our Intel dataset contains 875 nodes and 15605 edges, and each edge is modeled as a two-component max-mixture with a null hypothesis. The additional cost of handling mixtures is quite small in comparison to the total computation time. Run times were computed using the Java-based `april.graph` library, which is slower than `g2o`, but exhibits the same scaling behavior as other methods.

#### 4.5 Basin of Convergence

A key issue in non-linear optimization methods is whether the globally optimal solution will be found, or whether the optimization process will get stuck in a local minimum. This is a function of the initial solution as well as the parameters of the problem. In this section, we describe the effects of these parameters on the robustness of our method, as well as an experiment to empirically evaluate the magnitude of these effects.

The effect of the initial solution is relatively straight-forward: some initial solutions provide a better path for the optimization system to follow. In high-noise cases, some initial solutions may be far from the desired solution and in a different basin of convergence, leading to a poor solution. We consider two initializations: 1) open-loop odometry (well-suited to online optimization) and 2) an approach which initializes each node relative to its oldest neighbor (a heuristic used by TORO [Grisetti et al., 2007] and implemented within `g2o` [Kummerle et al., 2011], most useful in batch processing).

The type of errors that occur also affect the robustness of the method. In this analysis, we consider four types of erroneous loop closures based on the Vertigo package [Sünderhauf, 2012]: 1) random errors, 2) locally clustered (but not mutually consistent) errors, 3) randomly grouped

errors in which the groups are mutually consistent (group size = 10), and 4) locally grouped errors (group size = 10).

The relative “strength” of the null hypothesis in comparison to the putative hypothesis also has an effect on the optimization. This strength can be described in terms of two parameters. The weight parameter ( $w$ ) is the mixing parameter associated with the null-hypothesis component. Larger values of  $w$  increase the likelihood of the null hypothesis and cause the system to reject more of the putative hypotheses. If  $w$  is too small, and the system will accept incorrect hypotheses.

The second parameter,  $s$ , is the scale factor used in generating the information matrix associated with the null hypothesis from the information matrix associated with the putative hypothesis. When  $s = 1$ , both mixture components are identical and thus no robustness from the method can be expected. Smaller values (closer to zero) of  $s$  yield null hypotheses that are less certain than the putative hypothesis. This is equivalent to increasing its covariance, which pushes more probability mass away from the mean. This not only allows the null-hypothesis to produce a higher probability explanation of observed data, but also results in less curvature in the cost function. As  $s$  gets smaller, the cost function becomes increasingly flat, decreasing any influence of the mixture on the posterior solution.

We explore the effect of these parameters in Fig. 12. Columns of the table represent the four different outlier generation strategies and rows represent different data sets and initializations (not all combinations are presented for space reasons). Within each cell, the parameters  $w$  and  $s$  are swept resulting in a two-dimensional grid of map scores. At each data point, a graph was constructed and solved using the max mixture method and the log of the mean squared error (evaluated with respect to ground truth) is plotted according to color.

The data in Fig. 12 shows graphically how to tune the free parameters  $w$  and  $s$  to maximize the quality of the resulting map. Across virtually all of the experiments, the best performance is generally found in the lower right corner of the parameter sweep. This area corresponds to null-hypotheses with relatively large weights and low-information (equivalently large covariances). However, it is also evident that the region of good performance (which we subjectively appraise to be mean squared errors less than about  $-1$ ) is quite large in almost all cases. From these results, we conclude that the method is robust across many orders of magnitude of  $w$  and  $s$ , and that in general,  $w$  should be made relatively close to 1 and  $s$  relatively close to zero.

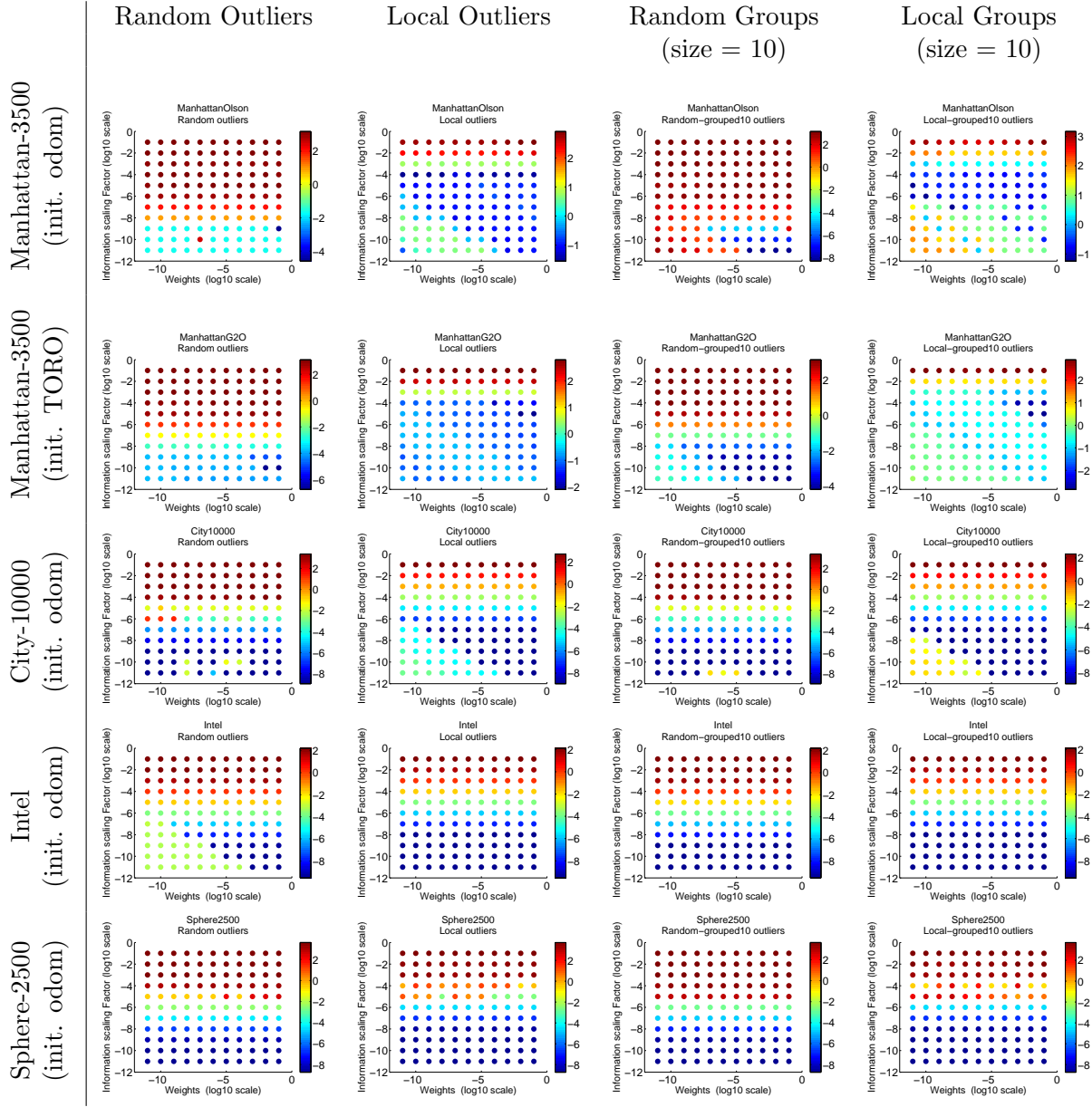


Figure 12: Robustness of method over a range of parameters. We consider five different data set+initial conditions (rows), four data association error generation methods (columns), and a parameter sweep over  $w$  and  $s$  within each grid cell. Colors correspond to the log of the mean squared error; maps less than  $-0.1$  are relatively good and maps less than  $-1$  are excellent. These plots show that the basin of convergence for the max mixture method is quite large. A total 1000 outliers of each error type was added for each dataset. For the grouped errors it resulted in 100 groups, each with 10 mutually consistent outliers.

## Acknowledgements

This research was partially supported by the U.S. Department of the Air Force, grant FA2386-11-1-4024, BMBF contract number 13EZ1129B-iView and European Commission under contract number ERC-267686-LifeNav.

## 5 Conclusion

We have described a method for performing inference on networks of mixtures, describing an application of our method to robot mapping. Our method consists of a novel mixture model based on a “max” operator that makes the computation of the Jacobian and residual fast, and we show how existing sparse factorization methods can be extended to incorporate these mixtures. We believe that such an approach is necessary for long-lived systems, since any system that relies on a *zero* error rate will fail.

We demonstrate how the mixture model allows null hypotheses and robustified cost functions to be incorporated into a maximum likelihood inference system. We show that our system is robust to a large error rates far in excess of what can be achieved with existing front-end loop validation methods. We further demonstrate a multi-modal formulation, addressing the “slip or grip” problem and showing that our system can make loop validation unnecessary in some cases.

Our algorithm cannot guarantee convergence to the global optimum, but we characterized the basin of convergence, demonstrating the relationship between error rate, node degree, and convergence to a good solution.

Finally, we demonstrate that the runtime performance of our algorithm is similar to that of existing state-of-the-art maximum likelihood systems. In comparison to other robust formulations, including those based on switching constraints, the ability of our method to encode “one-of-k” mixtures provides a significant performance advantage. Further, while we have explored the case of batch solvers, our method can be equally-well adapted to incremental systems [Kaess et al., 2008]. An open source implementation can be downloaded from [Agarwal et al., 2012]

## References

[Agarwal and Olson, 2012] Agarwal, P. and Olson, E. (2012). Variable reordering strategies for slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Sys-*



tems (IROS).

- [Agarwal et al., 2012] Agarwal, P., Olson, E., and Burgard, W. (2012). Max-mixture - open source implementation with g2o. <http://openslam.org/maxmixture.html>.
- [Bailey, 2002] Bailey, T. (2002). *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, Australian Centre for Field Robotics, University of Sydney.
- [Bailey et al., 2006] Bailey, T., Nieto, J., and Nebot, E. (2006). Consistency of the FastSLAM algorithm. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 424–429. Ieee.
- [Blackman, 2004] Blackman, S. S. (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19:5–18.
- [Bosse et al., 2004] Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *International Journal of Robotics Research*, 23(12):1113–1139.
- [Crossman et al., 2012] Crossman, J., Marinier, R., and Olson, E. (2012). A hands-off, multi-robot display for communicating situation awareness to operators. In *Proceedings of the International Conference on Collaboration Technologies and Systems*, pages 109–116.
- [Dellaert, 2005] Dellaert, F. (2005). Square root SAM. In *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, USA.
- [Dellaert and Kaess, 2006] Dellaert, F. and Kaess, M. (2006). Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1203.
- [Durrant-Whyte et al., 2003] Durrant-Whyte, H., Majumder, S., Thrun, S., de Battista, M., and Scheduling, S. (2003). A bayesian algorithm for simultaneous localisation and map building. In Jarvis, R. and Zelinsky, A., editors, *Robotics Research*, volume 6 of *Springer Tracts in Advanced Robotics*, pages 49–60. Springer Berlin / Heidelberg.
- [Eustice et al., 2006] Eustice, R., Singh, H., Leonard, J., and Walter, M. (2006). Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. *International Journal of Robotics Research*, 25(12):1223–1242.

- [Frese, 2005] Frese, U. (2005). A proof for the approximate sparsity of SLAM information matrices. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 331–337, Barcelona, Spain.
- [Grisetti et al., 2005] Grisetti, G., Stachniss, C., and Burgard, W. (2005). Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2432–2437, Barcelona.
- [Grisetti et al., 2007] Grisetti, G., Stachniss, C., Grzonka, S., and Burgard, W. (2007). A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA.
- [Hähnel et al., 2003] Hähnel, D., Burgard, W., Fox, D., and Thrun, S. (2003). A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 206–211.
- [Hartley and Zisserman, 2004] Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.
- [Howard and Roy, 2003] Howard, A. and Roy, N. (2003). The robotics data set repository (radish).
- [Kaess et al., 2008] Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Trans. on Robotics*, 24(6):1365–1378.
- [Konolige, 2010] Konolige, K. (2010). Sparse sparse bundle adjustment. In *British Machine Vision Conference*, Aberystwyth, Wales.
- [Kummerle et al., 2011] Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g2o: A general framework for graph optimization. In *ICRA*, Shanghai.
- [Kwak et al., 2007] Kwak, N., Kim, I.-K., Lee, H.-C., and Lee, B. H. (2007). Analysis of resampling process for the particle depletion problem in fastslam. In *IEEE International Symposium on Robots and Human Interactive Communications (RO-MAN)*, pages 200–205.

- [Latif et al., 2012a] Latif, Y., Cadena, C., and Neira, J. (2012a). Realizing, reversing, recovering: Incremental robust loop closing over time using the irrr algorithm. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4211–4217. IEEE.
- [Latif et al., 2012b] Latif, Y., Lerma, C. C., and Neira, J. (2012b). Robust loop closing over time. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia.
- [Lerner and Parr, 2001] Lerner, U. and Parr, R. (2001). Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proceedings of the Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 310–331, San Francisco, CA. Morgan Kaufmann.
- [Montemerlo, 2003] Montemerlo, M. (2003). *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- [Neira and Tardos, 2001] Neira, J. and Tardos, J. D. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897.
- [Newman, 1999] Newman, P. M. (1999). *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, University of Sydney, Australia.
- [Olson, 2008] Olson, E. (2008). *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.
- [Olson, 2009a] Olson, E. (2009a). Real-time correlative scan matching. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4387–4393, Kobe, Japan.
- [Olson, 2009b] Olson, E. (2009b). Recognizing places using spectrally clustered local matches. *Robotics and Autonomous Systems*, 57(12):1157–1172.
- [Olson, 2011] Olson, E. (2011). Evaluating back-ends: Metrics. In *Automated SLAM Evaluation Workshop, Robotics Science and Systems*, Los Angeles, USA.
- [Olson et al., 2012] Olson, E., Strom, J., Morton, R., Richardson, A., Ranganathan, P., Goeddel, R., Bulic, M., Crossman, J., and Marinier, B. (2012). Progress towards multi-robot reconnaissance and the MAGIC 2010 competition. *Journal of Field Robotics*, 29(5):762–792.

- [Pfingsthorn and Birk, 2012] Pfingsthorn, M. and Birk, A. (2012). Simultaneous localization and mapping (slam) with multimodal probability distributions. *International Journal of Robotics Research (IJRR)*.
- [Ranganathan et al., 2010] Ranganathan, P., Morton, R., Richardson, A., Strom, J., Goeddel, R., Bulic, M., and Olson, E. (2010). Coordinating a team of robots for urban reconnaissance. In *Proceedings of the Land Warfare Conference (LWC)*.
- [Sibley et al., 2009] Sibley, G., Mei, C., Reid, I., and Newman, P. (2009). Adaptive relative bundle adjustment. In *Robotics Science and Systems Conference*, Seattle, USA.
- [Smith et al., 1988] Smith, R., Self, M., and Cheeseman, P. (1988). A stochastic map for uncertain spatial relationships. In Faugeras, O. and Giralt, G., editors, *Proceedings of the International Symposium of Robotics Research (ISRR)*, pages 467–474.
- [Stachniss et al., 2005] Stachniss, C., Grisetti, G., and Burgard, W. (2005). Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 667–672, Barcelona, Spain.
- [Strasdat et al., 2010] Strasdat, H., Montiel, J., and Davison, A. (2010). Scale drift-aware large scale monocular slam. *Proceedings of Robotics: Science and Systems (RSS)*, 2(3):5.
- [Sünderhauf, 2012] Sünderhauf, N. (2012). Vertigo: Versatile extensions for robust inference using graphical models.
- [Sunderhauf and Protzel, 2012] Sunderhauf, N. and Protzel, P. (2012). Switchable constraints for robust pose graph slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [Thrun and Liu, 2003] Thrun, S. and Liu, Y. (2003). Multi-robot SLAM with sparse extended information filters. In *Proceedings of the International Symposium of Robotics Research (ISRR)*, Sienna, Italy.
- [Walter et al., 2005] Walter, M., Eustice, R., and Leonard, J. (2005). A provably consistent method for imposing exact sparsity in feature-based SLAM information filters. In Thrun, S., Brooks, R., and Durrant-Whyte, H., editors, *Proceedings of the International Symposium of Robotics Research (ISRR)*, pages 214–234, San Francisco, CA. Springer.

# Learning Convolutional Filters for Interest Point Detection

Andrew Richardson   Edwin Olson

**Abstract**— We present a method for learning efficient feature detectors based on in-situ evaluation as an alternative to hand-engineered feature detection methods. We demonstrate our in-situ learning approach by developing a feature detector optimized for stereo visual odometry.

Our feature detector parameterization is that of a convolutional filter. We show that feature detectors competitive with the best hand-designed alternatives can be learned by random sampling in the space of convolutional filters and we provide a way to bias the search toward regions of the search space that produce effective results. Further, we describe our approach for obtaining the ground-truth data needed by our learning system in real, everyday environments.

## I. INTRODUCTION

In this paper, we present a method to automatically learn feature detectors that perform as well as the best hand-designed alternatives and are tailored to the desired application. Most current feature detectors are hand-designed. Creating feature detectors by hand allows the designer to leverage human intuition to create intricate and efficient detectors that would be hard to replicate automatically. However, many feature detectors tend to be reused across multiple applications with subtle differences that could be leveraged to improve performance. Automatically-learned detectors have the potential to exploit these differences and improve application performance, as well as help us better understand the properties of top-performing feature detectors. Such a learning process, however, comes with its own challenges, such as defining a sufficiently general parameterization in which good detectors can be found and tractably exploring such a parameter space.

This work approaches automatic feature detector learning by learning fast and effective feature detectors based on *convolutional filters*. Convolutional filters make up an expressive space of feature detectors yet possess favorable computational properties. We specifically learn these detectors for use in stereo Visual Odometry (VO), an application in which the full set of feature detector invariances, such as affine invariance, are not required to achieve good motion estimates and efficiency is key. Unlike feature detection and matching evaluations limited to planar scenes, these detectors are learned on realistic video sequences in everyday environments to ensure the relevance of the learned results. Learning high-performance feature detectors also requires a good source of training data, which can be difficult or expensive to obtain. We detail our method for generating

The authors are with the Computer Science and Engineering Department, University of Michigan, Ann Arbor, MI 48104, USA {chardson,ebolson}@umich.edu  
<http://april.eecs.umich.edu>

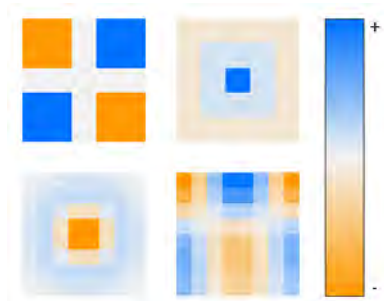


Fig. 1: When using convolutional feature detectors, which filter is best? Reasonable examples include the corner and point detectors used in [8] (top, both) and a Difference-of-Gaussians (DOG) filter (bottom left) like those used for scale-space search in SIFT [12]. We automatically generate filters for feature detection to improve the 3D motion estimation of a stereo visual odometry system. Bottom right: the most accurate filter in this work.

ground truth data — instrumenting an environment with 2D fiducial markers known as AprilTags [17]. These markers can be robustly detected with low false-positive rates, allowing us to extract features with known data association across an entire video. We can then solve for the global poses of all cameras and AprilTags and use this information to evaluate the motion estimate when using natural feature detections. Importantly, we also use this global reconstruction to reject any natural feature detections on or near AprilTags, ensuring that we do not bias the learning process to detect fiducial markers.

The feature detectors learned with our method are efficient and accurate. Processing times are comparable to FAST, a well-known method for feature detection, and reprojection errors for stereo visual odometry are lower than those with FAST in most cases [18]. In addition, because our filters use a simple convolutional structure, processing times are reduced by both increases in CPU clock rates and SIMD vector instruction widths.

The main contributions of this paper are:

- 1) We propose a framework for learning a feature detector designed to maximize performance of a specific application.
- 2) We propose convolutional filters as a family of feature detectors with good computational properties for general-purpose vector instruction hardware (e.g. SSE, NEON).
- 3) We present a sampling-based search algorithm that can incorporate empirical evidence that suggests where high

quality detectors can be found and tolerate both a large search space and a noisy objective function. We detail both the search algorithm and a set of experiments used to steer the search towards effective portions of the search space.

- 4) We present a method to evaluate the performance of learned filters with easily-collected ground truth data. This enables evaluation of the end application in arbitrary 3D environments.

In Section II, we discuss background material and prior work. We then discuss the general principles of our method for detector learning in Section III. Section IV focuses on the specifics of our application, stereo visual odometry, and ground-truthing method. Finally, our experiments and evaluation are presented in Section V.

## II. BACKGROUND

Many feature detectors have been designed to enhance feature matching repeatability and accuracy through properties such as rotation, scale, lighting, and affine-warp invariance. Some well known examples include SIFT [12], SURF [2], Harris [9], and FAST [18]. While SIFT and SURF aim to solve the scale-invariant feature detection problem, Harris and FAST detect single-scale point features with rotation invariance at high framerates. Other detectors aim to also achieve affine-warp invariance to better cope with the effects of viewpoint changes [14].

Many comparative evaluations of feature detectors and descriptors are present in the literature [14], [16]. Breaking from previous research, Gauglitz et al. evaluated detectors and descriptors specifically for monocular visual tracking tasks on video streams [7]. This evaluation is beneficial, as continuous motion between sequential frames can limit the range of distortions, such as changes in rotation, scale, or lighting, that the feature detectors and descriptors must handle, especially in comparison to image-based search methods that can make no such assumptions. Our performance-analysis mechanism is similar; however, whereas [7] focused on rotation, scale, and lighting changes for visual tracking, we focus on non-planar 3D scenes.

An alternative to hand-crafted feature detectors and descriptors is automated improvement through machine learning. FAST, which enforces a brightness constraint on a segment of a Bresenham circle via a decision tree, is a hand-designed detector that has been optimized for efficiency via ID3 [18]. An extension of FAST, FAST-ER, used simulated annealing to maximize detection repeatability [19]. In contrast to these approaches, we focus on learning a detector to improve the output of our target application using a method with a low and nearly-constant feature detection time.

Detector-learning work by Trujillo and Olague used genetic programming to assemble feature detectors composed of primitive operations, such as Gaussian blurring and equalization [20]. Their results are promising, though the training dataset size was small. Additionally, they attempt to maximize detector repeatability, whereas our method is

focused on the end-to-end system performance of our target application.

In addition to the detector-learning methods, descriptor learning methods like those from Brown et al. learn local image descriptors to improve matching performance [4]. They use discriminative classification and a ground-truthed 3D dataset. Their resulting descriptors perform significantly better than SIFT on the ROC curve, even with shorter descriptors. As this paper focuses on *detector* learning, we use a common *descriptor* for all detectors. This descriptor uses a standard pixel-patch representation and allows an even comparison between all detectors evaluated in this work.

## III. LEARNING A FEATURE DETECTOR

Feature detector learning requires three main components: a parameterization for the detector, an evaluation metric, and a learning algorithm. The parameterization defines a continuum of detectors, ideally capable of describing the range from fixed-size point or corner detectors to scale-invariant blob detectors, as well as concepts like “zero mean” filters. The evaluation harness computes the error of a proposed detector, which we want to minimize. Given these components, we can construct a method to generate feature detectors that maximize our learning objective. While iterative optimization through gradient or coordinate descent are popular ways to solve such problems, these approaches are problematic in learning feature detectors due to the high-dimensional search space and noise in the objective function. Random sampling allows us to evaluate far more filters than with iterative methods, find good filters despite numerous local minima, and develop an intuition for the constraints on the filter design that yield the best performance.

### A. Detector parameterization

We parameterize our feature detector as a convolutional filter [15]. This is an attractive representation due to the convolutional filter’s flexibility and the ability to leverage signal processing theory to interpret or constrain the qualities of a detector. In addition to a convolutional filter’s flexibility, these filters can also be implemented very efficiently on vector instruction hardware, such as Intel SSE, AVX, and ARM NEON. This hardware is commonly available in modern smartphone processors, as rich media applications can benefit significantly from SIMD parallelism.

We want to find the convolutional filter that yields the most accurate result for our target application. This is different from the standard metrics for feature detector evaluation like repeatability, as the best features may not in fact be detectable under all rotations. Our objective function (which we minimize) measures the error in the motion estimate against ground truth. This is in contrast to methods which maximize an approximation of end-to-end performance like repeatability [19], [20]. The advantage of our approach is the potential to exploit properties specific to the application. In stereo visual odometry, for example, edges that are vertical from the perspective of the camera are easy to match between

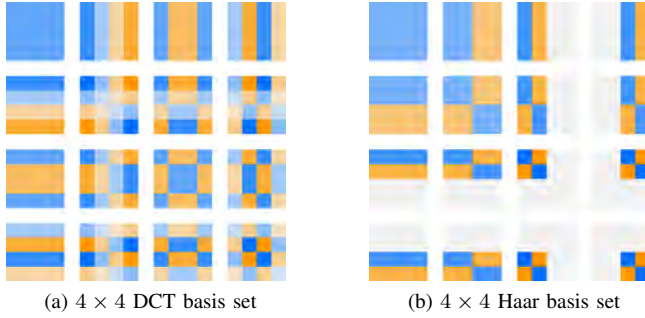


Fig. 2: Basis sets representing each frequency-domain transform for (a) the Discrete Cosine Transform (DCT) and (b) the Haar Wavelet Transform on a  $4 \times 4$  image patch. Each patch corresponds to a single coefficient in frequency space. The top left basis corresponds to the DC component of the filter. Note that we use  $8 \times 8$  filters in this work.

the left and right frames due to the epipolar geometry constraints. This property is not captured by standard measures like repeatability, so methods which use these measures cannot be expected to exploit them.

Our detection method can be summarized as follows:

- 1) Convolve with the filter to compute the image response
- 2) Detect points exceeding the filter response threshold, which is updated at runtime to detect a constant user-specified number of features
- 3) Apply non-extrema suppression using the filter response over a  $3 \times 3$  window

This work focuses on  $8 \times 8$  convolutional filters. A naïve parameterization would simply specify the value of each entry in the filter, a space of size  $\mathbb{R}^{64}$ . In the following section, we detail alternative parameterizations which allow us to learn filters which both perform better and require less time to learn.

### B. Frequency domain parameterizations

Within the general class of convolutional filters, we parameterize our feature detectors with frequency domain representations. In this way, we can apply meaningful constraints on the qualities of these filters that would not be easily specified in the spatial domain. In doing so, we learn about the important characteristics for successful feature detectors built from convolutional filters.

We use the Discrete Cosine Transform (DCT) and Haar Wavelet transform to describe our filters [1], [15]. Unlike the Fast Fourier Transform (FFT), the DCT and Haar Wavelets only use real-valued coefficients and are known to represent image data more compactly than the FFT [3]. This compactness is often exploited in image compression and allows us to sample candidate filters more efficiently. These transformations can be easily represented by orthonormal matrices and computed through linear matrix products.

In this work, we make use of three representations for filters — a straightforward pixel representation, the Discrete Cosine Transform (DCT), and the Haar Wavelet Transform.

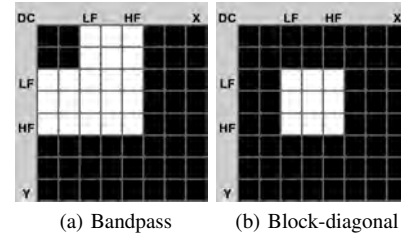


Fig. 3: Example frequency-domain filter constraints for  $8 \times 8$  filters. Coefficients rendered in black are suppressed (forced to zero). White coefficients can take any value. A typical bandpass filter is shown in (a). We propose the use of the block-diagonal region of support in (b), which exhibited superior performance in our tests.

Figure 2 shows a set of basis patches for the two frequency-domain representations. The pixel values of a filter in the spatial domain can be uniquely described by a weighted linear combination of these basis patches, where the weights are the *coefficients* determined by each frequency transformation. For convenience, we refer to both the spatial values of the filters (pixel values) and frequency coefficients as  $\mathbf{w}$  for the remainder of this paper.

### C. Error minimization

In our application, our goal is to find a convolutional filter which yields the best motion estimate for a stereo VO system. We represent the error function that evaluates the accuracy of a proposed filter by  $E(\mathbf{w})$ . As described further in Section IV-B, our error function is the mean reprojection error of the ground truth data, the four corners of the AprilTags, using the known camera calibrations and the camera motion estimated using the detector under evaluation.

While iterative optimization via gradient or coordinate descent methods is a straightforward approach for learning with an error function, we found that such iterative methods get caught in local minima too frequently. We propose instead to learn detectors by randomly-sampling frequency coefficient values while varying the size and position of a coefficient mask that zeroes all coefficient values outside of the mask. We refer to our mask of choice as a block-diagonal constraint, as illustrated in Figure 3. We also evaluate the performance of sampling with bandpass constraints and sampling raw pixel values via the naïve approach. In all cases, sampled values are taken from a uniformly-random distribution in the range  $[-127, 127]$ .

The constraints illustrated in Figure 3 are defined by low and high frequency cutoffs, which define the filter’s bandwidth. The filters shown have low and high frequency cutoffs of 0.250 and 0.625, respectively<sup>1</sup>. Thus, the filters have a bandwidth of 0.375.

Iteratively-optimizing filters can be very expensive. Steepest descent methods that use the local gradient of the error function require at least  $n$  error evaluations for square filters

<sup>1</sup>Note that we use frequency ranges normalized to the interval  $(0, 1)$



of width  $\sqrt{n}$ . After gradient calculation, multiple step lengths may be tried in a line-search minimization algorithm. At a minimum,  $n + 1$  error evaluations are required for every update. Coordinate descent methods require at least  $2n$  evaluations to update every coefficient once. Both methods require more calculations in practice. Because the error surface contains a high number of local minima, step sizes must be small and optimization converges quickly. This results in a great deal of computation for only small changes to the filter. We found that 95% of our best randomly-sampled filters did not reduce their error significantly after iterative optimization. Many did not improve at all.

In contrast to iterative methods, computing the error for a new filter only requires one evaluation. The result is that in the time it would take to perform one round of gradient descent for an  $8 \times 8$  filter, we can evaluate a minimum of 65 random samples.

#### IV. STEREO VISUAL ODOMETRY

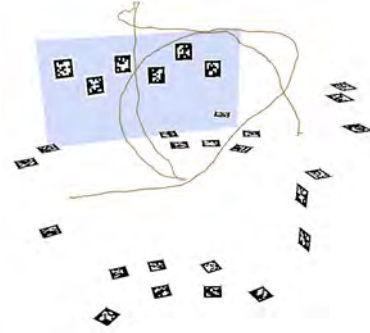
In this section, we describe aspects of our application domain, stereo visual odometry. In principle, our in-situ training methods apply to other stereo visual odometry pipelines and other applications. We use a custom stereo visual odometry pipeline for feature detector learning. Prior work in this area has yielded accurate and reliable results with high-framerate VO using corner detectors [13], [11]. A number of system architectures are possible and have been demonstrated in the literature, including monocular [11], [5] and stereo approaches [13], [8].

##### A. Visual Odometry overview

Our approach to stereo visual odometry can be divided into a number of sequential steps:

- 1) **Image acquisition** - 30 Hz hardware-triggered frames are paired using embedded frame counters before stereo rectification via bilinear interpolation.
- 2) **Feature detection** - Features are detected in grayscale images with non-extrema suppression. Zero-mean,  $9 \times 9$  pixel patch descriptors are used for all features.
- 3) **Matching** - Features are matched between paired images using a zero-mean Sum of Absolute Differences (SAD) error score. To increase robustness to noise, we search over a  $\pm 1$  pixel offset when matching. Unique matches are triangulated and added to the map. Previously-mapped features are projected using their last known 3D position and matched locally.
- 4) **Outlier rejection** - We provide robustness to bad matches with both Random Sample Consensus (RANSAC) and robustified cost functions during optimization (specifically, the Cauchy cost function with  $b = 1.0$ ) [6], [10].
- 5) **Motion estimation** - We initialize the motion estimate to the best pose from RANSAC. We then use nonlinear optimization to improve the point positions and camera motion estimates, iterating until convergence.

The result is an updated 3D feature set and an estimate of the camera motion between the two sequential updates.



(a) Reconstructed ground truth trajectory



(b) Reprojected ground truth features

Fig. 4: Ground truth reconstruction using AprilTags. Stereo camera trajectory (orange) in (a) is reconstructed using interest points set on the tag corners determined by the tag detection algorithm. Shaded region (blue) corresponds to the scene viewed in (b). The mask overlays (red) in (b) are the result of reprojecting the tag corners using the ground truth reconstruction and are used to ensure that no features are detected on the AprilTags added to the scene. Example feature detections from the best filter learned in this work are shown for reference (green).

##### B. Ground truth using AprilTags

We compute our ground truth camera motion by instrumenting the scene with AprilTags and solving a global non-linear optimization over all of the tags and camera positions. Specifically, we treat the four tag corners as point features with unique IDs for global data association. During learning, we explicitly reject any feature detections on top of or within a small window around any AprilTag so that we do not bias the detector learning process. In other words, our system rejects detections that would otherwise occur because of the AprilTags in the scene so that we learn to make use of the *natural* features in the environment. This is illustrated in Figure 4b, where red overlays correspond to regions where all feature detections are discarded. By using the reprojected AprilTag positions, we can reject features on AprilTags even when a tag is not detectable in the current frame.

Once the ground truth has been computed, we can compute the error of a motion estimate computed with an arbitrary feature detector. For every sequential pair of poses in the dataset, we compute the 3D position of the AprilTag corners with respect to the first of the two poses, transform from the first to the second pose's reference frame with the motion estimate from the arbitrary feature detector, and compute the reprojection error of these points in the images from the

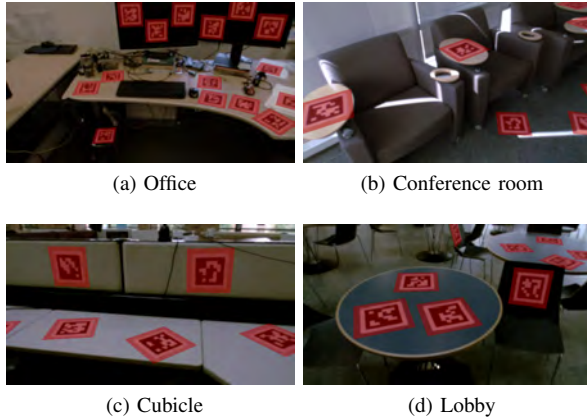


Fig. 5: Images from the four datasets used in this work. AprilTag masks described previously are shown in red.

second pose.  $E(w)$  is the mean reprojection error over all pairs of poses in the dataset using this method. It measures how well, on average, the ground truth features are aligned with their observations when using the candidate detector.

## V. EXPERIMENTS

Our experiments focus on randomly-sampling filters under different constraints and computing the mean reprojection error when using these filters across multiple datasets. In addition, we compare both error and computation time to existing and widely-used feature detectors.

### A. Implementation Details

Training and testing take place on four datasets between 23 and 56 seconds in length with 30 FPS video. These datasets were collected in various indoor environments, including an office, conference room, cubicle, and lobby. In each case, we randomly selected 15 seconds of video for training. Figures 4 and 5 show camera trajectories and imagery from these datasets. Our stereo rig uses two Point Grey FireFly MV USB2.0 color cameras at a resolution of  $376 \times 240$ . Experiments were run on a pair of 12-core 2.5 GHz Intel Xeon servers, each with 32 GB of memory. Sampling 5,000 filters takes approximately 7 hours at 9.7 seconds per filter.

In contrast to the substantial computational resources used in learning, our target application is limited to the compute available on a typical mobile robot. A mobile-grade processor such as the OMAP4460, a dual-core ARM Cortex-A9 device, used as an image preprocessing board, is a compelling and scalable computing solution to our needs. With a convolution implementation optimized via vector instructions, specifically ARM NEON, we can detect around 300 features per  $376 \times 240$  image in 3.65 ms for a  $8 \times 8$  filter. In comparison, the ID3-optimized version of the FAST feature detector performs similarly, requiring 3.20 ms. For both methods, we dynamically adjust the detection threshold to ensure the desired number of features are detected even as the environment changes.

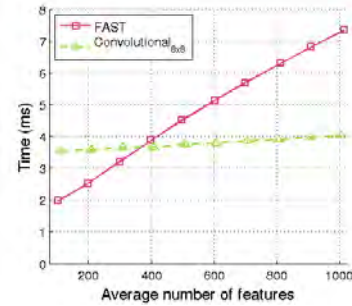


Fig. 6: Time comparison between FAST-9 and an  $8 \times 8$  convolutional filter feature detector. Times represent the combined detection and non-extrema suppression time and were computed on the PandaBoard ES (OMAP 4460) over 30 seconds of video with  $376 \times 240$ , grayscale imagery.

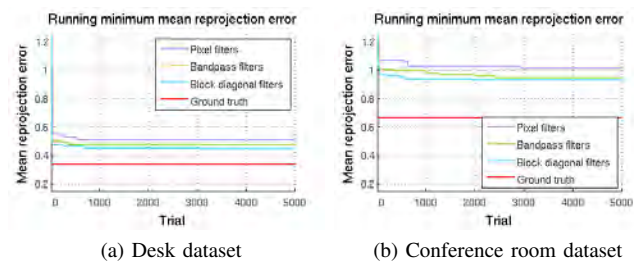


Fig. 7: Mean reprojection error for the best filter sampled so far (running minimum error) over 5,000 samples. Ground truth system error shown in red.

While both methods are efficient enough for real-time use, the difference in computation time for a large number of features is dramatic. Figure 6 shows the runtime as a function of the desired number of features for both methods. This time measurement includes detection and non-extrema suppression. While both methods show a linear growth in computation time, the growth for the convolutional methods is much slower than for FAST. This is because the convolution time does not change as the detection threshold is reduced. The linear growth is due only to non-extrema suppression. This result is especially important for methods where the desired number of detections is high [11], [8].

### B. Randomly-sampled filters

We evaluated our approach by running a suite of random-sampling experiments for block-diagonal filters with both the DCT and Haar parameterization. We also compare to sampled bandpass and pixel filters. Figure 7 shows the error of the best filter sampled so far as 5,000 filters are sampled. In all four datasets, the pixel and bandpass filters never outperformed the best block-diagonal filter. The final filters of each type and from each dataset<sup>2</sup> are shown in Figure 8.

Figure 9 shows the error distributions for each of the three constraints on the conference room dataset. For each experiment, 5,000 filters were randomly generated with the

<sup>2</sup>Final coefficients available at [umich.edu/~chardson/icra2013feature.html](http://umich.edu/~chardson/icra2013feature.html)

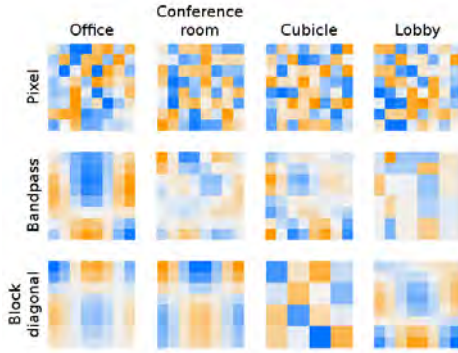


Fig. 8: Best learned filter for every type and dataset combination. Best viewed in color.

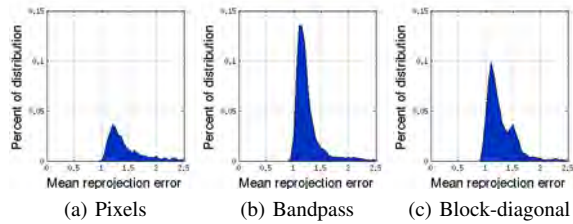
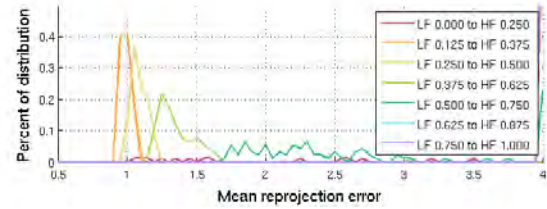


Fig. 9: Histogram of mean reprojection errors for three filter generation methods on the conference room dataset. While randomly-generated bandpass filters yield better performance, on average, than filters with uniformly-random pixel values, block-diagonal filters have both better average performance and a lower error for the best filters. 79% and 72% of sampled bandpass and block-diagonal filters, respectively, had errors under 2.5 pixels, while only 30% of random pixel filters had such low errors.

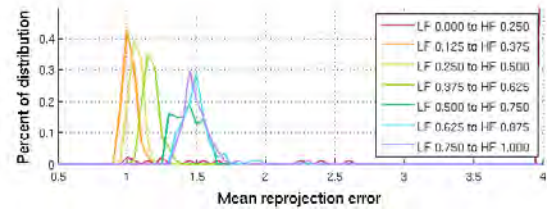
appropriate set of constraints. For the bandpass and block-diagonal constraints, we generated filters for all combinations of the DCT or Haar transform, low frequency cutoff and filter bandwidth, defined previously. Of the 27 combinations of low and high-frequency mask cutoffs available for both DCT and Haar filters of size  $8 \times 8$  (in total,  $\binom{8}{2} - 1$  combinations each for DCT and Haar), only 6 of them include the DC component and, in the case of the block-diagonal filters, the vertical and horizontal edge components.

From these plots, it is clear that limiting the search for a good filter through the bandpass and block-diagonal sampling constraints significantly improved the percentage of filters which yield low reprojection errors. Our interpretation of this result is that filters are very sensitive to nonzero values for specific frequency components. By strictly removing these components in 21 of the 27 constraint combinations, the average filter performance improves significantly.

Figure 10 shows separate distributions for block-diagonal filters for each of the possible low-frequency cutoffs for filters with the most narrow filter bandwidth (0.250). One plot is shown for each frequency transformation (DCT and Haar). From these figures, it is clear that filters perform



(a) Distributions for sampled DCT block-diagonal filters



(b) Distributions for sampled Haar block-diagonal filters

Fig. 10: Distributions for block-diagonal filters with a bandwidth of 0.250 as a function of the low frequency cutoff on the conference room dataset. For both (a) and (b), the filters which include the DC and vertical/horizontal edge components (LF cutoff of 0) have reprojection errors greater than 4 pixels 84% and 87% of the time for DCT and Haar filters, respectively. Beyond the DC components, only the DCT filters with low-frequency cutoff of 0.5 or greater have such large reprojection errors. The remaining distributions progress smoothly from low error (left) to high error (right) as the frequency increases. Best viewed in color.

significantly worse when the DC and edge coefficient values are not zero. Beyond that, we see a trend of low error for low-frequency filters, and an increasing error as frequency increases. Finally, the DCT filters seriously degrade when they begin to contain high frequency components; however, the Haar filters do not. Our interpretation of this result is that, because Haar basis patches are not periodic, a simple step transition in an image will often result in a unique maxima. In contrast, the periodic DCT basis patches will yield multiple local maxima, causing a cluster of detections around edges. Similar trends exist for filters with higher bandwidths.

The performance of the best sampled block-diagonal filters is compared to baseline methods such as FAST, Shi-Tomasi, a Difference of Gaussians filter, and filters used by Geiger et al in Table I. The best result from every testing dataset (row) is shown in bold. All detector evaluations were performed with the same system parameters: detect 300 features after non-extrema suppression and filtering with the AprilTag masks, use RANSAC and a Cauchy robust cost function ( $b = 1.0$ ), etc. These parameters were set via parameter sweeps using the FAST feature detector on the office and conference room dataset. As such, these represent best-case conditions for FAST. Mean values over 25 trials are reported due to variability induced by RANSAC. The differences in the means between the trained filters and FAST were statistically significant in 10 of the 12 cases with p values

Testing dataset	Filters trained on specified dataset				Linear baseline methods			Nonlinear baseline methods		
	Office	Conf rm.	Cubicle	Lobby	DOG	Geiger Corner	Geiger Blob	FAST	Shi-Tomasi	SURF
Office	0.447	<b>0.466</b>	0.471	0.468	40.281 <sup>+</sup>	0.492	0.595	0.470	2.060	1.322*
Conf rm.	0.981	0.929	0.996	0.981	1.505	1.047	1.218	<b>0.953</b>	1.141	1.584*
Cubicle	1.292	<b>1.142</b>	1.134	1.368	2.962	2.131	4.042	1.441	4.550	0.795*
Lobby	1.593	<b>1.552</b>	1.628	1.482	1.974	1.573	1.927	1.654	1.938	2.032*

TABLE I: Testing error on each dataset using the learned feature detectors and baseline methods. Reported numbers are mean values over 25 trials to compensate for the variability of RANSAC, except for the training errors (gray). Bolded values are the best result for every row. \*SURF generated features adjacent to AprilTags that could not be easily filtered out because of SURF’s scale invariance. As such, the SURF results are not considered a fair comparison to the other methods. <sup>+</sup>Large errors are the result of data association failures with the specified features.

less than 0.01 for a two-tailed t-test.

These results reinforce the notion that learned convolutional filters can compete with nonlinear detection methods, like the FAST feature detector. Only on the conference room dataset did FAST perform better than a learned filter. On average, learned filters had a lower reprojection error than FAST by a small amount, 0.05 pixels. For other baseline methods, such as Shi-Tomasi, the improvement in reprojection error was substantial. Note that while SURF outperformed all methods on the cubicle datasets, this is due to SURF detections adjacent to AprilTags that cannot be rejected as easily due to SURF’s scale invariance.

For the linear baseline methods, the results vary greatly. Geiger et al’s corner filter performs the best of the three, and yet it and the other linear baselines perform quite poorly on the cubicle dataset, unlike the learned filters. Interestingly, these linear detectors (or an equivalent  $8 \times 8$  filter) are simply a few of the convolutional filters that could have been learned in our framework.

These results also suggest that dataset choice, not learned filter, was the best predictor of testing errors. The cubicle dataset had a high error in a number of cases. From inspecting the video stream, this is not surprising — the cubicle is generally featureless except for a few smooth edges and a narrow strip at the top where the camera sees over the cubicle wall.

## VI. SUMMARY

We have presented a framework for automatically learning feature detectors that can be efficiently computed on modern architectures and result in performance that is generally better than existing methods, sometimes substantially so. By sweeping over frequency-domain constraints on the filters during sampling, we learn detectors that outperform obvious alternatives and prior work. In addition, our results indicate that the best detectors are typically those that respond primarily to the lowest non-DC frequency components. These learned detectors perform well on all datasets, despite only using one dataset during training.

## ACKNOWLEDGEMENTS

This work was supported by U.S. DoD Grant FA2386-11-1-4024.

## REFERENCES

- [1] N. Ahmed, T. Natarajan, and K. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, 100(1):90–93, 1974.
- [2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Computer Vision—ECCV 2006*, pages 404–417, 2006.
- [3] T. Bose, F. Meyer, and M. Chen. *Digital signal and image processing*. J. Wiley, 2004.
- [4] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE transactions on pattern analysis and machine intelligence*, pages 43–57, 2010.
- [5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1052–1067, 2007.
- [6] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.
- [7] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, pages 1–26, 2011.
- [8] A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.
- [9] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
- [10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [11] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’07)*, Nara, Japan, November 2007.
- [12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.
- [13] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, pages 1–17, 2010.
- [14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1):43–72, 2005.
- [15] T. Moon and W. Stirling. *Mathematical methods and algorithms for signal processing*, volume 204. Prentice hall, 2000.
- [16] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 800 – 807 Vol. 1, 2005.
- [17] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.
- [18] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *Computer Vision—ECCV 2006*, pages 430–443, 2006.
- [19] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 2008.
- [20] L. Trujillo and G. Olague. Automated design of image operators that detect interest points. *Evolutionary Computation*, 16(4):483–507, 2008.



# AprilCal: Assisted and repeatable camera calibration

Andrew Richardson   Johannes Strom   Edwin Olson

**Abstract**—Reliable and accurate camera calibration usually requires an expert intuition to reliably constrain all of the parameters in the camera model. Existing toolboxes ask users to capture images of a calibration target in positions of their choosing, after which the maximum-likelihood calibration is computed using all images in a batch optimization. We introduce a new interactive methodology that uses the current calibration state to suggest the position of the target in the next image and to verify that the final model parameters meet the accuracy requirements specified by the user.

Suggesting target positions relies on the ability to score candidate suggestions and their effect on the calibration. We describe two methods for scoring target positions: one that computes the stability of the focal length estimates for initializing the calibration, and another that subsequently quantifies the model uncertainty in pixel space.

We demonstrate that our resulting system, AprilCal, consistently yields more accurate camera calibrations than standard tools using results from a set of human trials. We also demonstrate that our approach is applicable for a variety of lenses.

## I. INTRODUCTION

Applications such as visual odometry [14], dense reconstruction [8], [15], and colored point cloud segmentation [20] are fundamentally dependent on accurate calibrations in order to extract metrical data from images. The MATLAB and OpenCV packages are two popular systems for calibrating lenses [3], [4]. However, they can be error prone, especially for lenses with significant distortion. This stems from the fact that the quality of a calibration is dramatically affected by the user’s choice of calibration images. A user who chooses poor calibration target positions may find the resulting model generalizes poorly to unseen examples. This challenge is particularly acute for novice users, who are not aware of the properties of the underlying estimation and optimization methods, or end-users in dramatically different fields [2]. Even experts may be unsure that the positions they have chosen will yield a sufficiently accurate calibration, as the number of images needed is not constant across lenses and should vary with the quality of the constraints. Consequently, standard practice is to collect many more images than necessary and verify that the model parameter uncertainty and training error are low; if the results are unsatisfactory, the calibration is repeated or updated with additional images. This process is unreliable, and not very satisfying from a theoretical standpoint.

Therefore, the primary goal of this work is to increase calibration repeatability and accuracy in a more principled

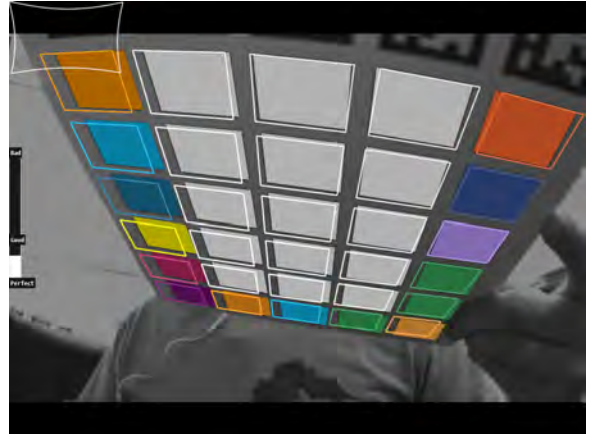


Fig. 1: The AprilCal GUI. Our system combines the ability to reason about unseen targets and a novel quality metric to make suggestions to the user about where to place the target. The user is notified that calibration is complete once the desired accuracy has been reached, typically achieving  $< 1$  pixel of error after 6-8 images.

fashion. We introduce a paradigm where fit quality is explicitly considered at each stage during a *live* calibration process. Specifically, we automatically consider many unseen target positions and suggest positions that will best improve the quality of the calibration. This is achieved using a novel quality metric based on the uncertainty of the calibration as measured in pixels. Previous toolboxes report the uncertainty of the *model* parameters, but the effect of these parameter uncertainties on pixel coordinates can be complex. We argue that worst-case uncertainty in pixels is more relevant for application performance and more natural for the user. Worst-case pixel uncertainty also serves as a principled basis to automatically determine when enough images have been collected.

We also introduce a new method for robustly bootstrapping a calibration that enables our system to make sensible recommendations even when little or no prior information is available about the lens. Our system also makes use of a calibration target composed of AprilTags [16], which, unlike previous approaches, can still be detected when individual markers are occluded. This enables a wider variety of target positions, which our method successfully exploits when making suggestions to the user.

We validated our camera calibration toolbox via a 16-participant study mostly comprised of users who had never calibrated a camera. Despite their lack of expertise, they were consistently able to use our software to produce

The authors are with the Computer Science and Engineering Department, University of Michigan, Ann Arbor, MI 48104, USA {chardson, jhstrom, ebolson}@umich.edu  
<http://april.eecs.umich.edu>

accurate calibrations. The same novice users also used the OpenCV calibration toolbox and invariably produced poorer calibrations, in some cases yielding errors of tens of pixels. In addition, we show that our toolbox can calibrate a wide range of lenses with an explicit accuracy guarantee.

The contributions of this work include:

- A framework for generating target position suggestions to guide users during camera calibration
- A new evaluation metric for camera calibration that enables high confidence in calibration parameters everywhere in the image
- A bootstrapping setup to enable interactive calibration even before all parameters are fully constrained
- An evaluation that demonstrates the robustness of our approach using a human-subjects study of 16 people, mostly novices.

AprilCal is released as part of the APRIL Robotics Toolkit and is available at <http://april.eecs.umich.edu>

## II. BACKGROUND

A wide variety of camera calibration approaches exist, spanning different optimization methods, calibration target styles and intrinsic model designs. Many previous methods have used multiple views of a planar target [22], [3], [4] or a single view of a carefully constructed 3D target [13]. Other methods have used laser pointers or other bright lights to facilitate calibration of networks of cameras. Such approaches typically still require bootstrapping by calibrating some cameras in the network with a constructed target [1], [21]. All of these prior methods share the same approach to calibration: a user first collects a set of images, then runs a batch calibration process on that data. This is in contrast to our approach, where the entire calibration process is interactive and additional data is solicited until the desired accuracy has been achieved.

A dominant paradigm for calibration involves capturing several images of a planar target. These approaches (ours among them) make associations between points detected in the image and corresponding world points on the target whose relative position are known by construction [22]. Simultaneous optimization over the intrinsic parameters for the camera model and the extrinsics for each target yield an estimate of the model parameters. Using such an approach requires the choice of 1) optimization method and 2) lens model.

Among the many possible optimization techniques, we adopt a standard, iterative non-linear-least-squares approach, using a sparse matrix solver as the back-end. This method is roughly analogous to standard approaches in GraphSLAM and bundle adjustment [5], [12], [11]. Our calibration vector  $x$  consists of all the model parameters (roughly 10) for the camera, in addition to the 6-DOF position of each calibration target. For each image containing  $k$  extracted 2D image points, we add  $2k$  linearized constraints as rows in the Jacobian matrix  $J$ . Each row-pair corresponds to projecting a feature from a known 3D coordinate on the calibration target into pixel coordinates, capturing both the unknown

position of the camera and the unknown camera parameters. Iterative solutions to Eqn 1 yield a locally-optimal set of model parameters for  $x$ .

$$J^T \Sigma_z^{-1} J \Delta x = J^T \Sigma_z^{-1} r \quad (1)$$

$$x_{i+1} = x_i + \Delta x \quad (2)$$

Here,  $\Sigma_z$  is the matrix of prior covariances for the target detector, and  $r$  is the residual, the observed minus the predicted pixel coordinates for each point. The correct convergence of  $x$  to the global minimum is sensitive to initialization of  $x_0$ ; we will discuss our approaches to this in Section III-A.

There are also a wide variety of models for camera intrinsics, starting with the fundamental pinhole model [7]. However, using the ideal pinhole model in isolation will poorly capture the dynamics of most real world lenses, especially those with a wide field of view. Therefore, many models extend this method by accounting for the lens distortion explicitly. For example, the MATLAB toolbox uses a polynomial Taylor series with 3-5 distortion terms to approximate these effects after projecting with the pinhole camera model [3]. In contrast, we have found that a polynomial as a function of  $\theta$ , the angle from the principle axis, yields as good or better<sup>1</sup> calibrations, often with fewer distortion terms for the lenses tested, increases the stability of the calibration process, and handles  $Z \leq 0$ . This is a reduced version of the model by Kannala and Brandt [10], which also includes tangential distortion.

The details of this angular polynomial model are shown in Equations 3-8, where  $X$ ,  $Y$ , and  $Z$  represent the 3D position of a point,  $\theta$  the angle from the principal axis,  $\psi$  the angle around the principal axis,  $x_{dn}$  the distorted point before converting to distorted pixel coordinates,  $x_{dp}$ , via the matrix  $K$ . The number of distortion coefficients is variable, though we use three to four in this work.

$$\theta = \arctan 2 \left( \sqrt{X^2 + Y^2}, Z \right) \quad (3)$$

$$\psi = \arctan 2 (Y, X) \quad (4)$$

$$r(\theta) = \theta + k_1 \theta^3 + k_2 \theta^5 + k_3 \theta^7 + k_4 \theta^9 + \dots \quad (5)$$

$$x_{dn} = [r(\theta) \cos(\psi), r(\theta) \sin(\psi)]^T \quad (6)$$

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$x_{dp} = K \cdot [x_{dn}, y_{dn}, 1]^T \quad (8)$$

In a similar spirit to this paper, several others have sought to make calibration easier, less time consuming and less error prone. For example, the ROS calibration package for the PR2 now has specific guidelines for the user about which checkerboard positions are required for getting a “good” calibration [18]. However, even with good rules of thumb, it is still possible that a user will collect “bad” frames that

<sup>1</sup>The details of this analysis are omitted due to space limitations. However, AprilCal can perform model selection to evaluate all available models as a post-processing step. See <http://april.eecs.umich.edu> for details.

lead to an inaccurate calibration. While it is possible to manipulate a calibration target automatically using a robotic arm [17], our approach can be used to choose desired target views during calibration. Others have shown that in some cases, it can be possible to calibrate a distorted camera using only a single image [2]. However, this approach does not explicitly constrain the accuracy of the resulting calibration, and works with only a very specific distortion model. In contrast, our method ensures that the user takes sufficient images to meet their desired level of accuracy, typically 6-8 images to achieve a  $< 1$  pixel confidence.

Finally, as suggested in Ranganathan’s work on non-parametric intrinsics models [19], we adopt the use of a strict testing set to provide a more rigorous evaluation of the actual accuracy of the proposed method. This is in contrast to the standard practice of reporting only the *training* error.

### III. PROPOSED METHOD

As outlined in Section I, our proposed method improves on the state of the art by offering a virtual calibration assistant that provides suggestions to users and automatically notifies them when the calibration has reached the specified accuracy. Our approach leverages a calibration target consisting of a mosaic of AprilTags [16], which can be detected robustly on the live video stream, even if parts of the target are occluded. Users interact with a GUI to match target positions suggested by our software. After each target position is achieved, the next best position is computed, repeating until the desired accuracy is achieved.

Our system divides the calibration into two sequential phases: bootstrapping, and uncertainty reduction. In the bootstrapping phase, we start with a restricted camera model with very few parameters, relaxing to the full model as new images of the target are added. Once all model parameters are fully constrained (typically after 3 images), we switch to the uncertainty reduction phase until the desired accuracy is achieved. Transition from the first to second phase is transparent: in both cases, the UI remains the same.

#### A. Bootstrapping a live calibration from scratch

Suggesting target positions is an inherently cyclic process: making good proposals that are actually realizable given the lens distortion requires a reasonably accurate calibration. However, a good calibration relies on having already captured several target positions to properly constrain all model parameters [22]. Therefore, to initialize the model parameters as quickly as possible, we initially use a reduced camera model; at the start of each calibration we assume that the focal center is at the center of the image and that there is no image distortion. This allows us to estimate the focal length after a single frame.

We can use this limited model immediately to choose the next-best target position, relaxing the reduced model to the full model, shown in Eqns. 3 - 8, as more frames are taken. This method is intended to select the target positions that best constrain the calibration while avoiding degenerate combinations [22]. We compute the calibration

initialization using a standard intrinsics matrix estimation technique – estimating the Image of the Absolute Conic (IAC) from perpendicular vanishing points and then decomposing it to estimate the intrinsics matrix [22], [9]. Using this initialization method, we score potential suggestions by sampling from the observation model to empirically compute the uncertainty of the intrinsics estimate. In other words, we prefer suggestions that yield intrinsics estimates with low variance. In our implementation, we estimate the focal length over 20 trials, each time adding uniformly-random, zero-mean noise to the image coordinates of the tag detections.

Some lenses generate too much distortion for IAC intrinsics matrix estimation. However, methods exist to remove the distortion from a single image [6], [2]. Such methods could be easily added to AprilCal, but this was unnecessary for the moderately distorted, wide field of view lenses tested in this work (see Figure 9).

Before the first suggestion can be shown to the user, we must obtain a cursory estimate of the camera calibration. We achieve this by automatically selecting the first image “behind the scenes” as the user moves the target to the center of the screen. However, the very first frame may not provide a robust initialization. To make this initialization robust, we score the live image stream and replace the first frame (removing the previous one) every time we find a new frame with a lower intrinsics uncertainty, either until a threshold is met, or the first suggestion has been computed and captured. This method reliably picks a satisfactory first frame because the user is guided to move the calibration target to a suggestion as soon as *any* frame has been captured and the intrinsics matrix has been estimated.

Once the calibration has been initialized, we can consider the effect of observing an unobserved frame on the uncertainty of the parameter estimates. For each candidate target position drawn from a coarse grid in pose space, we score the intrinsics estimate resulting from the combination of 1) the frames acquired so far (ignoring initial frames that were replaced) and 2) the projection of the candidate calibration target with the current estimates of the calibration parameters. As before, we sample from the observation model to estimate the uncertainty of the intrinsic parameters, choosing the suggestion that reduces the parameter variance the most.

In addition to providing full-rank constraints for all parameters of the complete camera model, the bootstrapping process also provides a good initialization for  $x_0$  in the optimization described in Eqn.1. As the model is successively relaxed, we pass through the initialization from the previous step, yielding good estimates for all the intrinsics parameters. Once the distortion parameters are introduced, we initialize them to zero. Given sane intrinsics estimates, these parameters converge well in practice.

#### B. Pixel-based calibration error metric

Once all intrinsic parameters are fully constrained, the next goal is to find enough additional target observations so that we are confident that the resulting model parameters are



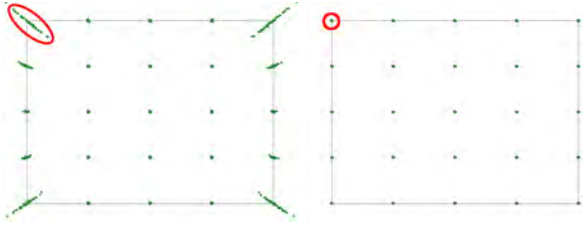


Fig. 2: Computing the Max Expected Reprojection Error (Max ERE) from the test points, and resulting error distributions after 3 and 5 images. Our sampling-based error metric can be used with any intrinsics model and allows us to automatically ensure the calibration is well constrained everywhere without requiring the user to collect a test set. The set of samples with the Max ERE are circled in red.

---

**Algorithm 1** COMPUTE\_MAX\_ERE(currentCalibration)

---

```

(x, Σ) = getModelPosterior(currentCalibration)
meanCal = makeCal(x)
testPointsXYZ = makeTestGrid(meanCal, 5, 5)
calSamples = [sampleCal0(x, Σ), ..., sampleCaln(x, Σ)]
MaxERE = 0
for all  $\vec{t} \in \text{testPointsXYZ}$  do
    ERE = 0
    for all sampCal  $\in$  calSamples do
        ERE +=  $\frac{1}{n} |\text{meanCal.project}(\vec{t}) - \text{sampCal.project}(\vec{t})|$ 
    end for
    MaxERE = max(ERE, MaxERE)
end for
return MaxERE

```

---

accurate. Previous approaches have used Mean Reprojection Error (MRE) and Mean Squared Error (MSE) as primary indicators of calibration quality. However, this is problematic because these are *training* errors, rather than testing errors. Using one of the prior works, if the training images are selected poorly, the resulting MRE could be low, yet the generalization performance (e.g. as measured on a test set) could be very poor. Unfortunately, collecting a proper testing set can be onerous – for our evaluation we use an expert-selected set of 60 or more images from all over the camera’s field of view. Especially for novice users, it is not reasonable to expect they would be able to collect a “good” testing set. Even for expert users, this process is time-consuming and requires careful attention.

Therefore, our approach is to derive a more principled estimate of the testing error that can be computed automatically given an intermediate state of the calibration. Our proposal is called “Max Expected Reprojection Error” (Max ERE), which we can compute at any stage during the calibration by sampling from the current posterior distribution over the model parameters. We then project a series of 3D points through each sampled calibration, producing a distribution of pixels for each test point, whose mean error is the “ERE”. Finally, we take the max of the EREs over all the test points. This ensures that we will properly weight the part of the

image where the model is currently the most uncertain. We choose the fixed 3D points carefully so that they will project into all parts of the image. Our current implementation uses a 5 x 5 grid of test points distributed so their projections will uniformly cover the image (see Figure 2). See Alg. 1 for an overview of the implementation.

Computation of the Max ERE uses the estimate of the marginal posterior covariance of the model parameters:  $\bar{P}(m|z_0, \dots, z_n)$ . This distribution is derived by first computing the joint distribution of the model parameters and each target extrinsics, given all the observations of those targets. Suppose we have collected  $n$  images of targets, then we can “marginalize-out” the target extrinsics:

$$\bar{P}(m|z_0, \dots, z_n) = \int_{T_0, \dots, T_n} P(m, T_0, \dots, T_n | z_0, \dots, z_n) \quad (9)$$

where  $m = \{f_x, f_y, \dots, k_1, \dots\}$ ,  $T_i$  is a 6DOF rigid-body transform, and  $z_i$  contains the x, y pixel locations of the centers of every AprilTag in image  $i$ . In practice, we assume the joint distribution in Eqn. 9 can be approximated as multivariate Gaussian:

$$N(x, \Sigma) = N \left( \begin{bmatrix} m \\ T_0 \\ \vdots \\ T_n \end{bmatrix}, \begin{bmatrix} \Sigma_{m,m} & \Sigma_{m,T_0} & \dots & \Sigma_{m,T_n} \\ \Sigma_{T_0,m} & \ddots & & \vdots \\ \vdots & & \ddots & \\ \Sigma_{T_n,m} & \dots & & \Sigma_{T_n,T_n} \end{bmatrix} \right) \quad (10)$$

This allows the marginal  $P(m|z_0, \dots) = N(m, \Sigma_{m,m})$  to be computed trivially by dropping the other rows and columns of the covariance matrix. Computation of  $\Sigma$  requires inverting the sparse  $\text{dim}(x)$  by  $\text{dim}(x)$  information matrix  $\mathcal{I}$ , which is derived from the observed target positions (similar to Eqn 1):

$$\mathcal{I} = J^T \Sigma_z^{-1} J \quad (11)$$

where each row of  $J$  is the linearized projection equation describing how a point on the target projects into the image, given the model parameters  $m$  and target position  $T_i$ . Crucially, this process depends on an estimate for the detector accuracy in pixels,  $\sigma_z$ , which must be known in advance. For AprilTag, we have empirically found the accuracy to be relatively constant across lenses with image width as a satisfactory predictor. Proper focus of the lens is assumed.

$$\sigma_z = 7 \times 10^{-5} \times \text{width} \quad (12)$$

Detector accuracy was fit independently for a number of camera configurations (see Figure 8) using 60+ image calibration datasets for each. The resulting accuracies were then used to compute the linear model in Eqn. 12.

#### IV. IMPLEMENTATION

AprilCal is implemented in Java and runs at 25 FPS with  $640 \times 480$  images on a quad-core Intel i7-3740QM @ 2.7GHz. Using a mosaic of AprilTags as our calibration target allows us to automatically detect the target at video rates, with processing time typically dominated by AprilTag

detection. We have found that rigid mounting at an office supply store is inexpensive and yields a target durable enough for many uses. In addition, we can detect and recognize individual tags without observing the entire target. This makes it possible to add constraints in the corners of images, even for highly distorted lenses. In the multi-camera case, this allows calibration of cameras with adjacent, but non-overlapping, fields of view.

In addition to target detection, our implementation requires significant CPU when determining the next suggestion. In our implementation, we score a fixed set of about 60 target positions regularly distributed throughout the field of view. This process depends on incorporating hypothetical observations into the calibration optimization framework, and then estimating the marginal distribution over the model parameters. As more images are acquired, and the size of the joint distribution grows, this can take up to 1 or 2 seconds. However, this scoring process only occurs a small number of times: once after each suggestion has been achieved by the user.

The AprilCal user interface is designed primarily to allow the user to correctly match the target positions suggested by the system. As shown in Figure 1, we use a set of unique colors to show how the desired target position (hollow rectangles) should be matched by the live detections (solid rectangles). The UI also automatically offers basic advice to the user via textual prompts about how to move the target to match the desired pose. When the calibration is deemed complete, the user is then automatically presented with the rectified video stream. This allows the user to qualitatively verify that the calibration is accurate, primarily by checking for straightness of projected lines.

## V. HUMAN TRIALS

We conducted a series tests with human subjects<sup>2</sup> to measure the effectiveness of AprilCal and to compare it to the widely used OpenCV method. Our user population consisted of undergraduate students at the University of Michigan. Only 3 of the 16 subjects reported any previous experience with camera calibration.

Our experiment protocol was as follows: each participant was asked to calibrate the same camera and medium-distortion lens with two different methods (see Figure 3). We used a Point Grey Chameleon CMLN-13S2M-CS in  $648 \times 482$  8-bit grayscale mode with a 2.8mm Tamron lens (Model 13FM28IR). This lens has a medium amount of distortion – significant enough that several Taylor series terms are required to model it, but still with a moderate field of view (only  $93^\circ$  horizontal FOV).

The two methods we evaluated were OpenCV’s calibration using automatic checkerboard detection and AprilCal using a mosaic of AprilTags. Participants were given a set of printed instructions. If they asked questions to the experimenter, they were given comprehension-level clarification on the



Fig. 3: Example images taken by the participants for both Method A (Open CV) and Method B (AprilCal).

instructions or advised to re-read the instructions. Participants then followed a checklist to first collect four samples using OpenCV, followed by four samples using AprilCal. Additionally, participants watched a video demonstrating calibration with each method. In Method A (OpenCV), participants interacted with a GUI showing live detections of the chessboard, using the “space” key on a wireless keyboard to capture a frame. In Method B (AprilCal), the frames are automatically taken when the participants move the targets close enough to the suggested pose.

In contrast to AprilCal, which provides detailed guidance throughout the calibration, OpenCV’s `calibrate.cpp` provides no in-application suggestions. Therefore, we designed a set of instructions for calibrating with their software. Our goal was primarily to emulate the experience of a first-time user who downloads this software from the Internet. Therefore, we provided users with some example pictures from the MATLAB Toolbox web page. The best written instructions we found were on the ROS tutorial for monocular camera calibration [18], which we also included. These are:

- checkerboard on the camera’s left, right top and bottom of field of view
- checkerboard at various sizes
- close (fill the whole view)
- far (fill  $\sim 1/5$  of the view)
- checkerboard tilted to the left, right top and bottom

After reading these instructions, participants were then instructed to take 10-16 images in each of the 4 trials (on the same lens).

## VI. EVALUATION

We evaluated AprilCal on several fronts. First, we report the results from our human subjects study to demonstrate the robustness of our approach to user error. Then, we demonstrate that our novel Max ERE metric is a good proxy for testing error. Finally, we demonstrate that AprilCal can be successfully used on a variety of lenses.

A note on evaluation of calibration quality: in all of our evaluations, we use testing error to indicate calibration quality. Each testing set is a collection of 60+ images from all over the field of view, including the corners of the images and at various scales. Because we do not have ground truth positions for the targets in the testing set, the error we

<sup>2</sup>Our study was reviewed by the University of Michigan Humanities Institutional Review Board and designated “exempt” with oversight number HUM00066852.

Dataset	Lens Model	Reprojection Error	
		Mean	Max
OpenCV	Radial, 3 distortion terms	0.728	38.646
AprilCal	Radial, 3 distortion terms	0.229	1.651
AprilCal	Angular, 4 distortion terms	0.203	1.444

TABLE I: Mean and max testing errors averaged over all human subject trials in comparison to a 65 image reference dataset for the same lens. Results are significant with  $p < 6.3 \times 10^{-7}$  for both mean and max errors and all pairs of rows.

report is after we optimize the target extrinsics to best fit the **fixed** model parameters for a given calibration. While this in general results in lower reprojection errors, it ensures that all models are fairly evaluated and still allows discrimination between good and bad calibrations.

Furthermore, Mean Reprojection Error (MRE) is typically reported as a summary of calibration quality, as it is simple to understand and robust to detector error. However, it can also often mask systematic errors in the underlying calibration. Therefore, we also report Max Reprojection Error on the test set – this ensures that calibrations are evaluated by their performance everywhere in the image.

#### A. Novice Calibrators using AprilCal and OpenCV

Our study results show that novices do a significantly better job calibrating when using AprilCal than when using OpenCV ( $p < 6.3 \times 10^{-7}$ ). For example, with testing set errors averaged over all participants, the testing MRE using OpenCV is approximately three times that when using AprilCal (see Table I). The disparity is even greater when considering the Max Reprojection Errors – OpenCV averages 38 pixels (6% of the image), whereas AprilCal averages a much lower 1.6 pixels for the same model. Interestingly, no OpenCV calibration yielded a max reprojection error better than the *worst* max reprojection error from AprilCal (2.02 pixels). This may be because the sorts of images that novice users capture, even when attempting to follow the ROS instructions, don’t constrain the whole lens well. With the target suggestions provided by AprilCal, even new users of camera calibration software can produce calibrations with very low worst-case reprojection errors. The error histograms for both populations is shown in Figure 4.

The human study results can also help us understand where in the image the calibrations disagree. Figure 6 depicts the expected error between the human trials and a 65 image reference calibration. From the images shown, it is clear that the OpenCV calibrations fail to capture the lens model in the image corners. This can be explained by both the need to observe the whole calibration target in OpenCV and the difficulty for users to predict where constraints are needed.

In addition to showing that AprilCal calibrations are more accurate, the user study results also show that calibrations with AprilCal are more consistent. Figure 5 depicts the distribution of focal lengths and focal centers for both AprilCal and OpenCV. While both distributions have similar means,

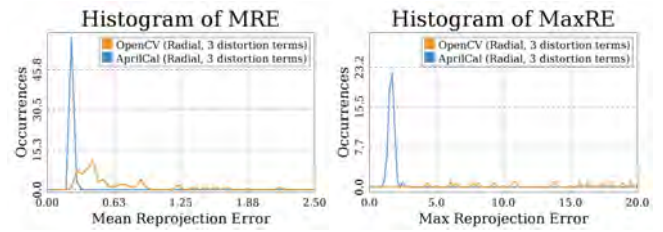
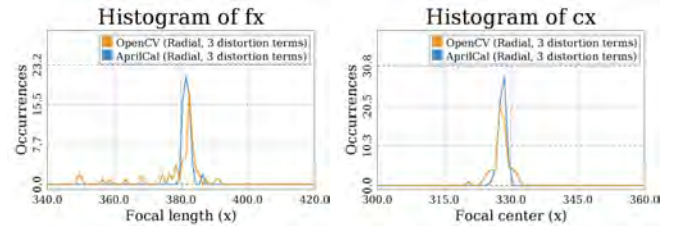


Fig. 4: Mean and Max Reprojection Errors (on a 65-image test set) for calibrations produced using AprilCal and OpenCV. Users produced significantly more reliable calibrations using the proposed method ( $p < 6.3 \times 10^{-7}$ ). For OpenCV, 3 MREs and 46 MaxREs not visible within plot extents.



Dataset	Focal length (x)		Focal center (x)	
	Mean	Std dev	Mean	Std dev
OpenCV	378.9 <sup>†</sup>	9.0 <sup>†</sup>	327.8 <sup>†</sup>	1.9 <sup>†</sup>
AprilCal	381.7	1.2	328.0	0.8

Fig. 5: Distribution of focal lengths and focal centers for all trials in the human study. While the mean parameter values from calibrations with AprilCal and OpenCV are similar, the standard deviations for the OpenCV calibrations are much higher. <sup>†</sup>One OpenCV outlier that would have further skewed the calculations was omitted from the calculations and is not visible within the plot extents. Best viewed in color.

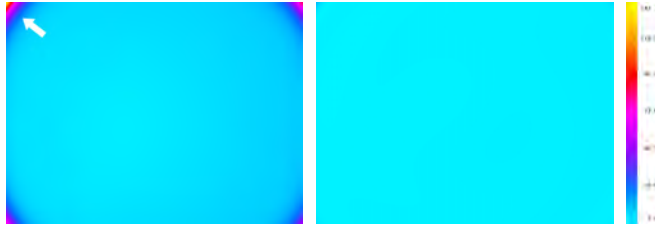
the AprilCal standard deviations are  $7.5\times$  smaller for focal lengths and  $2.37\times$  smaller for focal centers.

#### B. Evaluating the Max ERE metric

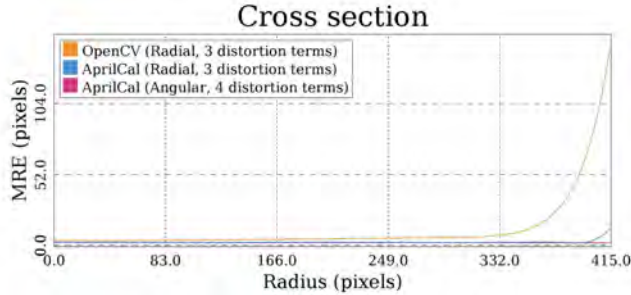
We designed the Max ERE to be a good measure of calibration quality. Specifically, we want users to specify the accuracy they need for their application (e.g.  $< 1$  px), and if the Max ERE falls below that threshold, then the calibration can confidently be said to be that accurate. To validate these claims, we computed several variants of testing error over a large number of AprilCal trials. After each image is added, we evaluate the performance using Max ERE, as well as on an independent testing set using Max (100th percentile), 99.5th percentile, and mean reprojection errors. As can be seen in Figure 7, our sampled-MRE metric corresponds closely to the highest percentiles of testing error. This shows empirical evidence that our metric is effective.

#### C. Accuracy of AprilCal on a Variety of Lenses

In addition to performing reliably for a wide range of users, AprilCal also produces accurate calibrations for a



(a) OpenCV (Radial, 3 dist. terms) (b) AprilCal (Radial, 3 dist. terms)



(d) Cross section from the focal center to the furthest corner in the image

Fig. 6: Heatmaps and cross section depicting per-pixel mean reprojection error between test subject calibrations and a 65 image testing set. Reprojection errors were calculated by projecting a point for every pixel in the reference calibration through all test subject calibrations, then computing the pixel distance. The reference calibration used an angular polynomial model with 4 terms, as it had the lowest mean and max training errors. AprilCal calibrations show low error in all parts of the image, while OpenCV calibrations have very high error in the corners (see arrow).

number of camera and lens configurations. Each lens was calibrated multiple times by one of the authors using the guidance provided by AprilCal (typically requiring 6-8 images in total). A separate 60+ image testing set was collected to evaluate the accuracy for each configuration. To fairly compare results from different lenses, we compute each lens' testing error against a reference calibration computed from the corresponding test set. This eliminates the effects of detector error on testing error, which varies for different images sizes (See Eqn 12). For each target point detected in the testing set, we project through both the reference calibration and the calibration using AprilCal, and compute the Mean, 99.5<sup>th</sup> percentile and Max Reprojections Errors. Figure 8 shows the testing error for six configurations that our lab uses for various robotics applications, including stereo odometry, object detection and overhead ground truth. In each case, the testing MRE is significantly below one pixel. Example images from each configuration are shown in Figure 9.

## VII. SUMMARY

AprilCal is an interactive calibration tool that provides live feedback on the state of the calibration and produces tightly-distributed calibration parameters even when used by

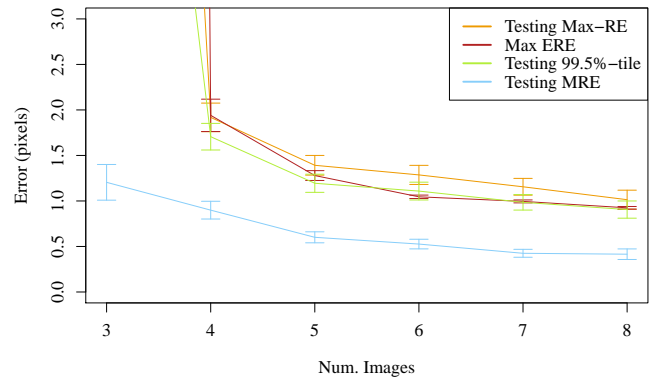
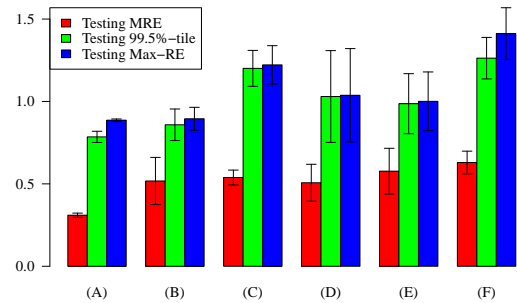


Fig. 7: Our novel Max Expected Reprojection Error (Max ERE) metric (red) correlates highly with the 99.5th percentile of reprojection errors on an independently captured test set. The Max ERE allows us to automatically compute a rigorous accuracy score for a partial calibration without needing an exhaustive test set. Error bars reflect std. error of the mean.



	Lens	DFOV	Resolution	Format
(a)	Fujinon YV2.2x1.4A-2	143°	648 × 482	Gray
(b)	Tamron 13FM22IR	146°	648 × 482	Gray
(c)	Tamron 13FM28IR	114°	648 × 482	Gray
(d)	Boowon BW38B	83°	752 × 480	Color
(e)	Boowon BW3M30B	121°	648 × 482	Gray
(f)	Boowon BW3M30B	121°	1296 × 964	Color

Fig. 8: Testing error for a variety of camera configurations using AprilCal. The Diagonal Field Of View (DFOV) was estimated from the testing sets for each configuration. Error bars reflect std. error of the mean.

novices. We have leveraged a novel calibration quality metric (Max ERE) to automatically determine whether a calibration is sufficiently accurate, without requiring a user to collect a rigorous testing set. We conducted a 16-person human subjects study to show that even novice users can produce consistent, quality calibrations using such a system.

We have evaluated AprilCal in a variety of ways, and demonstrated that it is a suitable replacement for the currently available calibration toolkits, which use a batch calibration process and training error as a quality metric. Our desire is to make accurate camera calibration available to a wider audience who can use the resulting model parameters confidently in a range of applications.



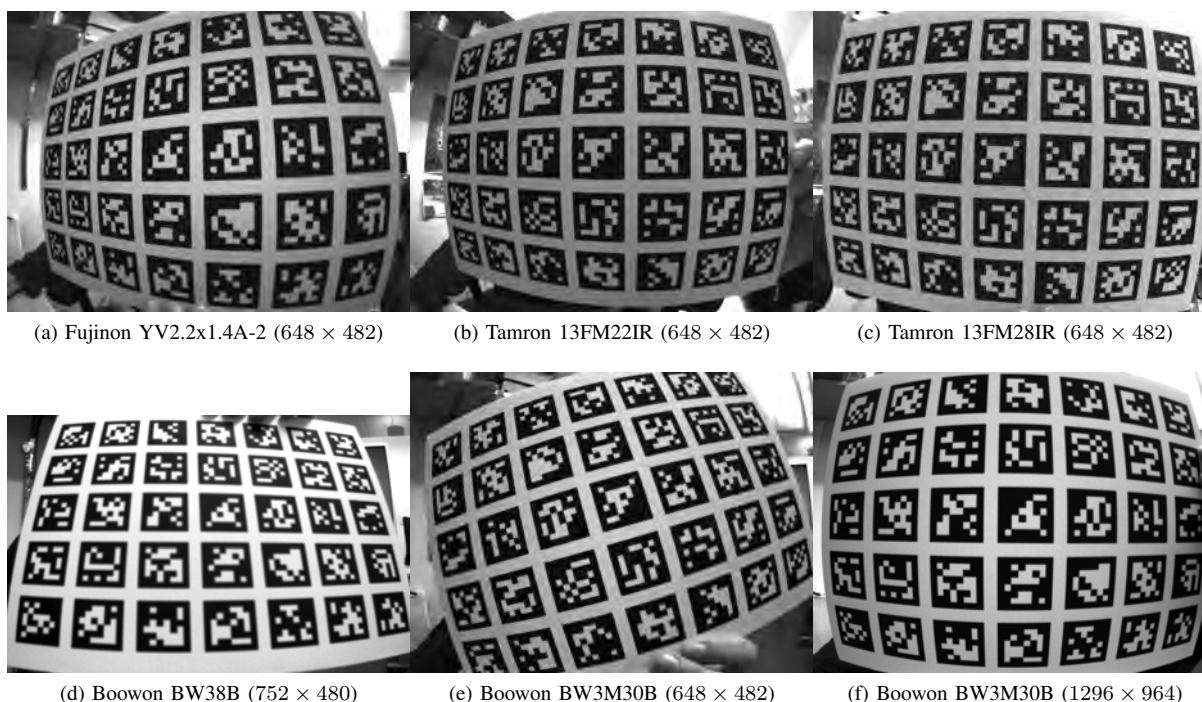


Fig. 9: Example images for each of the configurations evaluated in Figure 8. These lenses are representative of those known to work with AprilCal. Image contrast adjusted for clarity.

#### ACKNOWLEDGMENTS

This work was funded by U.S. DoD Grant FA2386-11-1-4024.

#### REFERENCES

- [1] J. Barreto and K. Daniilidis. Wide area multiple camera calibration and estimation of radial distortion. In *Omnivis-2004, ECCV-2004 workshop*, 2004.
- [2] J. Barreto, J. Roquette, P. Sturm, F. Fonseca, et al. Automatic camera calibration applied to medical endoscopy. In *20th British Machine Vision Conference (BMVC'09)*, 2009.
- [3] J.-Y. Bouguet. Camera calibration toolbox for MATLAB, July 2010.
- [4] G. Bradski. The OpenCV Library. *Dr. Dobbs' Journal of Software Tools*, 2000.
- [5] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, December 2006.
- [6] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.
- [7] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.
- [8] A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.
- [9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [10] J. Kannala and S. S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(8):1335–1340, 2006.
- [11] K. Konolige. Sparse sparse bundle adjustment. In *British Machine Vision Conference*, Aberystwyth, Wales, August 2010.
- [12] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *ICRA*, Shanghai, May 2011.
- [13] M. Li. Camera calibration of a head-eye system for active vision. *Computer Vision/ECCV'94*, pages 541–554, 1994.
- [14] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, pages 1–17, 2010.
- [15] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1498–1505. IEEE, 2010.
- [16] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.
- [17] V. Pradeep, K. Konolige, and E. Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *International Symposium on Experimental Robotics (ISER)*, New Delhi, India, 12/2010 2010.
- [18] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 2009.
- [19] P. Ranganathan and E. Olson. Gaussian process for lens distortion modeling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012.
- [20] J. Strom, A. Richardson, and E. Olson. Graph-based segmentation for colored 3D laser point clouds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2010.
- [21] T. Svoboda. A software for complete calibration of multicamera systems. In *Electronic Imaging 2005*, pages 115–128. International Society for Optics and Photonics, 2005.
- [22] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

# Robust Sensor Characterization via Max-Mixture Models: GPS Sensors

Ryan Morton and Edwin Olson

**Abstract**—Large position errors plague GNSS-based sensors (e.g., GPS) due to poor satellite configuration and multipath effects, resulting in frequent outliers. Due to quadratic cost functions when optimizing SLAM via nonlinear least square methods, a single such outlier can cause severe map distortions. Following in the footsteps of recent improvements in the robustness of SLAM optimization process, this work presents a framework for improving sensor noise characterizations by combining a machine learning approach with max-mixture error models. By using max-mixtures, the sensor’s noise distribution can be modeled to a desired accuracy, with robustness to outliers. We apply the framework to the task of accurately modeling the uncertainties of consumer-grade GPS sensors. Our method estimates the observation covariances using only weighted feature vectors and a single max operator, learning parameters off-line for efficient on-line calculation.

## I. INTRODUCTION

Most state-of-the-art Simultaneous Localization and Mapping (SLAM) algorithms require sensor uncertainties to be characterized probabilistically. Such characterization is a straightforward task for many common robot sensors, e.g., LIDARs. However, the noise distribution of other sensors provide a much more challenging characterization task. The cause of these erroneous observations stems from the inability of the sensor to observe the complete state of the environment. For example, ground robots’ wheels lose traction resulting in a slip-or-grip problem and vertical structures block Global Positioning System (GPS) signals, leading to multipath effects. Generally, the sensor return is assumed to reflect the most likely state of the quantity being measured, e.g., latitude and longitude for a GPS sensor<sup>1</sup>. The task of defining the uncertainty of these individual observations is termed sensor characterization.

The typical approach to sensor characterization uses Maximum Likelihood (ML) optimization to find the parameters that best describe the training data. This off-line process returns parameters that maximize the average performance over ground-truthed training data. These parameters can then be used on-line to calculate observation noise estimates for use within SLAM.

A single outlier can cause a severe, even irrecoverable, map distortion due to the quadratic cost surface used by the nonlinear least squares optimization within many SLAM algorithms. In this work, we take the view that outliers arise from overly-simple and optimistic noise models; a better noise model would assign a higher probability to

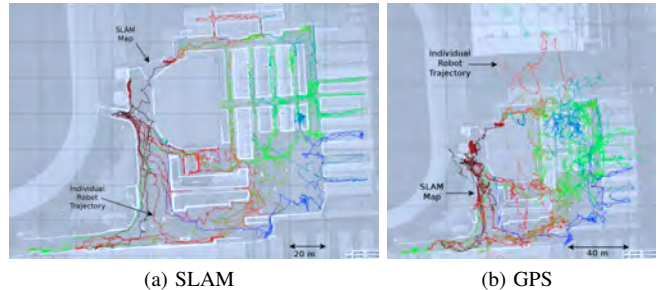


Fig. 1: Robot Trajectories for 12 Exploration Robots. Colored to reflect elapsing time (red-green-blue as  $t$  increases).

an “outlier”, thus limiting its effect. Simple Gaussian error models often suffer from this problem, since the tails drop off exponentially fast. We use max-mixtures of Gaussians to improve the richness of the sensor characterization models.

To highlight our framework, we show noise models for GPS sensors, which are particularly prone to large errors; it is common to receive observations from consumer grade sensors that are tens to hundreds of meters from the true location. Specifically, the contributions of this paper include:

- A machine-learning approach for sensor uncertainty estimation.
- An extension of sensor uncertainty estimation using max-mixtures.
- A new metric for evaluating the impact of outliers on a SLAM system.
- Evaluation of a variety of models for consumer-grade GPS sensor characterization.

In the next section, we give a short overview of related advancements in SLAM, robust estimation, and GPS error characterization. We describe the mathematical formulation of sensor characterization in Section III and show the robustness improvements of the max-mixture models in Section IV. Then in Section V, we describe features from GPS sensor data and how to combine them to robustly estimate GPS uncertainty. In Section VI, we show empirical results for the noise models on a real-world dataset comprised of 45 hours of GPS data from a 14-robot team.

## II. BACKGROUND

Many modern approaches formulate SLAM in terms of two components: a front-end that builds a factor graph and a back-end that optimizes it [1]. For example, the front-end might add an odometry constraint (edge) representing a rigid-body-transformation between two sequential robot positions

The authors are with the Computer Science and Engineering, University of Michigan, 2260 Hayward Street, Ann Arbor, Michigan, 48109, USA. {rmorton, ebolson}@umich.edu.

<sup>1</sup>In this work GPS means any Global Navigation Satellite System (GNSS)

(nodes); the constraint's mean represents the raw encoder observation while the edge weight is a function of the a priori uncertainty of the encoder. The back-end periodically optimizes the graph, producing a ML estimate for all nodes in the graph, i.e., the best position of the nodes given all the constraints. When errors are Gaussian this optimization can be viewed as a least squares method with each constraint incurring a quadratic cost.

A known problem in SLAM and other nonlinear least squares optimization problems stems from these quadratic costs, namely that a single outlier can have significant and detrimental impact on the solution [2], [3], [4]. The quadratic costs come directly from assuming constraints have Gaussian error, which brings computational efficiency.

Improving the robustness of SLAM to these outliers is currently an active area of research. Some approaches use switching variables, which can 'turn off' constraints and, thus, handle outliers by modifying the graph structure in the back-end [2], [5]. Another related approach detects and rejects outliers via loop consistency checks on small clusters of nodes [6]. Both of these approaches show promise at *dealing with outliers* in the back-end of SLAM. In this paper, we take the view that outliers arise from modeling errors and that richer error models can result in systems that are robust to outliers and do not require specialized "outlier rejection" methods.

While mixture models have long been used to approximate complex distributions, they typically result in high computational costs. Our system uses max-mixtures, which allows both flexible error models and fast inference [4]. Some sensors could be characterized with robust cost functions, [7], [8], but these have been previously shown to be subsumed by max-mixtures [4].

This paper focuses on models for GPS data, which can be used both to improve the accuracy of the map and to register the relative frame to Earth. Even with many loop closures, SLAM maps can have significant distortion compared to ground truth, e.g., long hallways may erroneously bend and GPS sensors can offer the needed constraints even when global registration is not necessarily needed [9]. Unmanned Aerial Vehicles (UAVs) and agricultural robots, which generally have unobstructed satellite line-of-sight, can rely heavily upon GPS sensors for positioning. In environments with an unobstructed view of the sky, GPS sensors offer cost-effective and computationally efficient position estimates that are, when supplemented with other sensors, sufficient for many systems [10]. Additional sensors, such as inertial measurement units or visual odometry have been shown to reduce errors caused by short blackouts in GPS data, i.e., obstruction of satellite view [11], [12], [13].

Early error models for GPS assumed a constant variance for all GPS returns [11], [14]. Many GPS sensors provide their own uncertainty estimates, but they can be misleading. For example, they may report erroneous readings for a period of time before detecting the loss of satellite locks. This problem stems from the discrete nature of individual satellite locks that can cause position estimate discontinuities [15], a

problem that is blamed for a crash during the DARPA Grand Challenge [16].

Rather than accurately modeling the noise of GPS sensors, a common approach has been binary classification of (in)valid sensor returns [13], [17], [16]. These systems stop adding GPS data to SLAM upon detection of a GPS dropout and/or multipath effects. Some approaches use additional sensors to improve multipath detection via satellite line-of-sight calculations by building 3D-models of the local structure [18], [19].

Some approaches circumvent the GPS sensor's position estimate and calculate positions directly from the trilateration of pseudo-ranges to individual satellites [20]. The most closely related work uses robust cost function and switch variables to modify the graph, but their method also requires a GPS sensor with pseudo-range capabilities [21]. Although these approaches using sensors equipped with pseudo-ranges and/or sensor utilizing the local structure have shown signs of success, we desire a robust approach that uses observations directly from standard consumer-grade GPS sensors.

### III. SENSOR CHARACTERIZATION

Sensor characterization consists of estimating the distribution of observations,  $\mathbf{z}_i$ , returned from the sensor around the true location,  $\mathbf{x}_i$ . The goal of sensor characterization is a distribution best defining the zero-mean  $P(\mathbf{z}_i|\mathbf{x}_i)$ . Designers generally choose the form of the distribution, e.g., Gaussian, and estimate the parameters of the model using training data, or by hand-tuning, to maximize log-likelihood. The output of the characterization is a covariance matrix  $\Sigma_i$ , which may be constant or a function of the observation.

#### A. Simple Gaussian Models

Let  $\mathbf{e}_i$  be the observation error, i.e.,  $\mathbf{e}_i = (\mathbf{z}_i - \mathbf{x}_i)$ . Assuming a zero-mean uni-modal Gaussian distribution, parametrized by  $\Sigma_i$ , the distribution becomes:

$$P(\mathbf{e}_i) = \mathcal{N}(\mathbf{0}, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2} \mathbf{e}_i^T \Sigma_i^{-1} \mathbf{e}_i} \quad (1)$$

with  $d$  equal to the degree-of-freedom (DOF) of the observation. The log-likelihood, assuming  $n$  independent observations, can be written:

$$\begin{aligned} L &= \log \prod_{i=0}^{n-1} P(\mathbf{e}_i) = \sum_{i=0}^{n-1} \log P(\mathbf{e}_i) \\ &= -\frac{nd}{2} \log(2\pi) - \frac{1}{2} \sum_{i=0}^{n-1} \log(|\Sigma_i|) + \mathbf{e}_i^T \Sigma_i^{-1} \mathbf{e}_i \end{aligned} \quad (2)$$

The last term in (2) explicitly shows the quadratic costs associated with deviations from the mean.

The sensor characterization task is the definition of  $\Sigma_i$ , a function of individual observations, to maximize  $L$ . Each such function can take many forms, but we use a parametrization where each element is a linear combination of features of the observation. For example, assuming a 2 DOF sensor



with independent noise in each DOF, the covariance may be defined as:

$$\Sigma_i = \begin{bmatrix} \sigma_{i,x}^2 & 0 \\ 0 & \sigma_{i,y}^2 \end{bmatrix} = \begin{bmatrix} (f_{i,x}^T w_x)^2 & 0 \\ 0 & (f_{i,y}^T w_y)^2 \end{bmatrix} \quad (3)$$

With feature vectors  $f_{i,x}$  and  $f_{i,y}$  as functions of each observation (presumably informing about the error in  $x$  and  $y$  respectively) and global weight vectors  $w_x$  and  $w_y$ . In this context, the characterization task would simply be the definition of the weight vectors.

#### IV. ROBUST SENSOR CHARACTERIZATION USING MAX-MIXTURES OF GAUSSIANS

Gaussian mixture models offer a richer representation by combining multiple Gaussian components, allowing modeling of arbitrarily complex distributions. Max-mixture models define the distribution as the max of  $k$  weighted components:

$$\begin{aligned} P(\mathbf{e}_i) &= \eta_i \max(\alpha_1 P_1(\mathbf{e}_i), \dots, \alpha_k P_k(\mathbf{e}_i)) \\ &= \eta_i \max_{j=1}^k \left( \frac{\alpha_j}{(2\pi)^{\frac{d}{2}} |\Sigma_{i,j}|^{\frac{1}{2}}} e^{-\frac{1}{2} \mathbf{e}_i^T \Sigma_{i,j}^{-1} \mathbf{e}_i} \right) \end{aligned} \quad (4)$$

Each components,  $P_j$ , is of the form in (1). The log-likelihood of the data becomes:

$$\begin{aligned} L &= \log \prod_i P(\mathbf{e}_i) = \sum_i \log P(\mathbf{e}_i) \\ &= \sum_i \log \left( \frac{\eta_i}{(2\pi)^{\frac{d}{2}}} \right) + \max_{j=1}^k \left( \log \left( \frac{\alpha_j e^{-\frac{1}{2} \mathbf{e}_i^T \Sigma_{i,j}^{-1} \mathbf{e}_i}}{|\Sigma_{i,j}|^{\frac{1}{2}}} \right) \right) \end{aligned} \quad (5)$$

Note that  $\eta_i$  becomes an additive constant that does not effect the minimization and does not generally need to be computed. The sensor characterization task for max-mixtures is to define the covariance estimates  $\Sigma_{i,j}$  and mixing terms  $\alpha_j$  that maximize  $L$ .

The most common mixture models are sum-mixtures, i.e.,  $P(\mathbf{e}_i) = \sum_{j=1}^k \alpha_j P_j(\mathbf{e}_i)$ , but our choice of max-mixtures is motivated by the use of logarithms in (5). The ability to push the logarithm inside the max operator brings significant computational advantages [4].

#### V. GPS SENSOR CHARACTERIZATION

We consider two families of GPS uncertainty predictors based on uni-modal Gaussians and max-mixtures of Gaussians using non-linear optimization for parameter fitting. The primary difference between these families is that, for each observation, uni-modal models have a single covariance estimate,  $\Sigma_i$ , and max-mixture models have  $k$  of them; plus  $k$  weight vectors.

For mobile ground robots, we assume radial GPS errors, with  $x$  and  $y$  errors independent and identically distributed, i.e.,  $e_i = \|\mathbf{e}_i\|$  represents the total translation error. Thus,  $f_i = f_{i,x} = f_{i,y}$ ,  $w = w_x = w_y$ , and each covariance function becomes:

$$\Sigma_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix} = \begin{bmatrix} (f_i^T w)^2 & 0 \\ 0 & (f_i^T w)^2 \end{bmatrix} \quad (6)$$

The global weights are learned off-line and we next describe some possible GPS feature vectors  $f_i$  for each observation  $z_i$ .

##### A. GPS Observation Features

An interesting property of GPS receivers is that they produce a wealth of data that can be used to generate features. In this section, we explore a variety of features, beginning with trivial (but still useful) ones and moving to more complex features.

1) *Constant Noise Model*: While few real-world systems today would attempt to characterize all GPS observations as having the same uncertainty, such a model can serve as a baseline method for evaluation. Using our framework, we simply let

$$f_i^T = [1]$$

Using this model, the magnitude of the learned weight would represent the standard deviation for all observations. Since all observations have equal feature vectors,  $f_i$ , all observations will have the same noise estimate  $f_i^T w$ .

2) *Number of Satellites Noise Model*: One strategy for determining the reliability of GPS data is to observe the number of visible satellites,  $n_{sat,i}$ . A simple feature vector expressing this assumption is given by:

$$f_i^T = [n_{sat,i}]$$

We expect to learn a negative weight for this feature since more satellites should reduce the observation error. However, this feature cannot be used alone, because negative  $\sigma$ s are prohibited. Later we'll see how these simple feature can be combined with others.

A more effective use of the number of satellites is to construct the feature vector  $f_i$  such that the  $n_{sat,i}^{th}$  element of  $f_i$  is set to 1, i.e., a "one-hot" encoding. For example, with 12 possible simultaneous satellite observations and a reported observation of 5 satellites, the feature vector would be given by:

$$f_i^T = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]$$

This representation allows a different covariance model to be fit to each number of satellites, but requires more training data through the whole range of possible  $n_{sat,i}$ . We expect to learn large weights for lower indices and small weights at the larger indices as the number of satellites should decrease the error.

3) *Dilution of Precision Noise Model*: A few standard outputs from GPS sensors are intended to represent the positional uncertainty in terms of geometric dilution of precision. Three such values are typically reported: horizontal dilution of precision (hdop), positional dilution of precision (pdop), and time dilution of precision (tdop). Each represents a multiplicative scaling of the uncertainty as a function of the geometric configuration of satellites, relative to the sensor, and should be informative of the true uncertainty. We can incorporate these values into our feature vector by letting (for example)  $f_i^T = [hdop_i]$ . We expect a positive correlation between these values and true uncertainty.

4) *Vendor-Provided Noise Model*: Many GPS units provide an uncertainty estimate computed by the sensor. To evaluate the quality of this estimate, we set our feature vector to contain just this estimate:

$$f_i^T = [ \sigma_{vendor,i} ]$$

If the vendor supplied estimates are correct, we would expect to learn a weight of 1 for this feature.

5) *Combination Noise Model*: The uncertainty estimate  $\sigma_i$  is a linear combination of features and, thus, the aforementioned features can be combined, with the individual feature and weight components retaining their original meaning. For example, combining the constant model, the simplified number of satellites model, and the vendor-provided estimate would produce the feature vector:

$$f_i^T = [ 1, n_{sat,i}, \sigma_{vendor,i} ]$$

We expect this feature to outperform the individual features, which are subsumed by the combination model, so long as over-fitting is avoided.

#### B. Additional Consideration for Max-Mixture Models

The aforementioned models can be used directly as uni-modal models of GPS uncertainty. However, simple Gaussian error models for GPS tend to perform poorly, as both earlier work and our experiments show. Yet, a  $k = 2$  max mixture model provides significant improvements. With  $k = 2$ , characterization of GPS sensors using max-mixtures requires maximizing  $L$  over a single  $\alpha$  value and two  $w$  vectors.

Let  $w_j$  be the dataset-wide weight vector associated with the  $j^{th}$  feature. For observation  $i$ , let  $f_{i,j}$  be feature  $j$  and standard deviation now given by  $\sigma_{i,j} = f_{i,j}^T w_j$ . For example, the features of a 2 component mixture of the constant model and the combination model (from previous section) are given by:

$$f_i = \begin{bmatrix} f_{i,1}^T \\ f_{i,2}^T \end{bmatrix} = \begin{bmatrix} [ 1 ] \\ [ 1, n_{sat,i}, \sigma_{vendor,i} ] \end{bmatrix} \quad (7)$$

With  $w$  now defining the  $k$  dataset-wide weight vectors  $w_j$ :

$$w = \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix} = \begin{bmatrix} [ w_{1,0} ] \\ [ w_{2,0}, w_{2,1}, w_{2,2} ] \end{bmatrix} \quad (8)$$

The feature vectors and weights combine, as expected to produce  $k$  standard deviations:

$$\begin{bmatrix} \sigma_{i,1} \\ \sigma_{i,2} \end{bmatrix} = \begin{bmatrix} f_{i,1}^T w_1 \\ f_{i,2}^T w_2 \end{bmatrix} \quad (9)$$

Combining with (5), this leads to a value for  $\sigma_i$  defined as:

$$\sigma_i = \arg \max_{\sigma_{i,j}} \left( \log(\alpha_j) - 2 \log(\sigma_{i,j}) - \frac{e_i^2}{2\sigma_{i,j}^2} \right) \quad (10)$$

To illustrate the max-mixture approach, suppose that some GPS measurements are nominal (with errors of a few meters), while other measurements are “outliers” (with errors of tens of meters). With a uni-modal approach using  $f = [ 1 ]$ , we might learn  $w = 9$ . However, by setting  $f = [ [ 1 ], [ 1 ] ]$  (a mixture of two constant variances) we might expect to

learn  $w = [ [ 2 ], [ 30 ] ]$  and a value of  $\alpha$  in relation to the frequency of those outliers. While simplistic we will show that this method works well.

## VI. EVALUATION

In this section we evaluate our proposed GPS noise models on a 14-robot dataset collected within a 220 x 160 m indoor/outdoor region of the Adelaide Showgrounds in South Australia using Garmin GPS18x-5Hz sensors during the MAGIC competition [22].

#### A. Performance Metrics

To analyze the GPS sensor characterization we analyze three primary metrics for each model: 1) likelihood of observed data, 2) robustness to high-error observations, and, more generally, 3) the distribution of error relative to model-predicted error.

We use the normalized log-likelihood of the data, given the model and parameters learned off-line, to measure the overall fit of the model. This metric represents the expected log-likelihood of each observation and we desire models that maximize this metric. Note that the normalization constant  $\eta$  is not computed, since it does not affect the maximization of this metric.

In the SLAM context, or any similar non-linear optimization problem, a single erroneous measurement can wreak havoc if not properly modeled. For uni-modal models, outliers have low probability and  $\min \log P(e_i)$  represents the worst-case likelihood error.

Since SLAM computes an ML solution via non-linear optimization and a 1<sup>st</sup> order Taylor expansion, the gradient magnitude,  $\|\nabla_i\|$ , is another outlier metric. During the optimization process  $\|\nabla_i\|$  represents the “pull” of the constraint, relative to other constraints’ gradients. For a given uni-modal Gaussian, the least likely measurement also has the maximum gradient, i.e.,  $\arg \min_i L_i = \arg \max_j \|\nabla_j\|$ . However, with max-mixtures an observation may be expected with near zero probability, but have virtually no gradient and thus no ‘pull’ within the optimization process. We desire models that minimize the “pull” associated with outlier measurements, i.e., we desire small  $\max \|\nabla_i\| = \max \|\Sigma_i e_i\|$ .

For each observation,  $e_i/\sigma_i$  represents the number of standard deviations predicted by the model. If the noise was truly distributed according to a uni-modal Gaussian distribution, the  $e_i/\sigma_i$  errors would be distributed as a  $\chi$  distribution. Although this metric does not exactly relate to max-mixtures, we still desire  $e_i/\sigma_i$  distributions that are quantitatively similar to a  $\chi$  distribution. For example, we hope to find models where all observations fall within the statistically significant region,  $\max(e_i/\sigma_i) < 6$ .

#### B. Analysis

Because our evaluation datasets were dominated by inliers, the normalized log likelihood of all of our models, including both simple Gaussian models and max-mixture models, falls within a relatively small range (from -7.3 to -6.3). These

TABLE I: Training and Testing Error.

Model	Uni-Modal						Max-Mixture					
	Train			Test			Train			Test		
	$\frac{1}{n} \sum L_i$	$\max \frac{e_i}{\sigma_i}$	$\max \ \nabla_i\ $	$\frac{1}{n} \sum L_i$	$\max \frac{e_i}{\sigma_i}$	$\max \ \nabla_i\ $	$\frac{1}{n} \sum L_i$	$\max \frac{e_i}{\sigma_i}$	$\max \ \nabla_i\ $	$\frac{1}{n} \sum L_i$	$\max \frac{e_i}{\sigma_i}$	$\max \ \nabla_i\ $
constant-hdop	-6.855	14.503	2.920	-6.852	17.895	2.750	-6.587	8.810	1.363	-6.623	10.946	1.300
constant-hdop-nsat-vendor	-6.299	6.866	2.260	-6.318	11.451	2.127	-6.289	5.142	1.840	-6.306	7.261	1.836
constant-hdop-vendor	-6.312	6.342	1.957	-6.325	11.548	1.890	-6.303	4.318	1.631	-6.325	7.932	1.664
constant-nsat-vendor	-6.302	6.785	2.208	-6.319	11.595	2.096	-6.271	6.736	2.431	-6.286	10.770	2.275
constant-vendor	-6.312	6.331	1.948	-6.326	11.516	1.878	-6.305	4.286	1.737	-6.321	7.292	1.791
constant	-7.210	21.106	3.263	-7.321	21.618	3.251	-6.733	12.353	1.118	-6.820	12.653	1.114
vendor	-6.329	6.177	2.514	-6.336	9.643	2.331	-6.320	4.204	2.048	-6.330	6.563	2.024

small differences arise from dramatic differences in the log likelihoods of the relatively-infrequent outlier measurements.

As seen in Table I, for any given feature vector, the performance of a mixture of two components invariably outperformed the best single component (simple Gaussian) error model, according to both training and test error. As measured by log likelihood, the magnitude of these differences is small, but again, this is due to the fact that the vast majority of the measurements were inliers that every model handled well. As expected, we find that test error is generally somewhat higher than training error, but the magnitude of the increase is similar between both the simple Gaussian model and the more complex max-mixture models.

Unlike the normalized log likelihood, in which the performance of the models on outliers is masked by the large number of inliers, the *worst case* standard deviation metric clearly shows the advantages of the mixture models. The worst-case standard deviation,  $\max e_i/\sigma_i$ , dropped for every model, often dramatically, e.g., from 21.618 to 12.653 in the constant-covariance model case. These improvements highlight a significant improvement in modeling the sensor’s noise. To be clear, measurements with 12 standard deviations of error may still have too much influence during optimization, but the constant model is a very naive baseline and ignores any uncertainty indications/features the sensor.

Histograms of the empirical  $\chi$  error compared to the model’s predicted density are also revealing (see Fig. 2). In the case of simple Gaussian models (left column), we see that the model is consistently conservative with respect to the inlier data (on the left side of the plots). The outliers have caused an increase in the covariance estimate, with the consequence that inliers are given too little weight. Despite the inflated covariance estimate, outliers still have a very high gradient (see Table I) due to their distance from the Gaussian distribution’s mean and will strongly influence the optimization.

Conversely, the right column of Fig. 2, which plots histograms of the max-mixture models, shows a model error that more tightly tracks the ideal distribution. Simultaneously, outliers are shifted closer to the left, indicating that higher probabilities have been predicted for them. These outliers still influence the graph, but have correspondingly smaller “pull”, which would make a SLAM system more resilient to them.

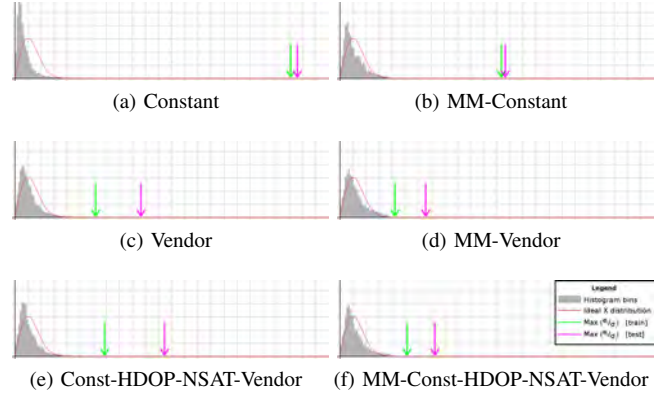


Fig. 2: Histogram of Empirical vs. Ideal  $\chi$ -error (for select models). Ideally, if the underlying Gaussian assumptions hold, the normalized histogram of  $e_i/\sigma_i$  would fit a 2 DOF  $\chi$  distribution (with horizontal units representing standard deviations,  $\sigma_i$ ). The relative movement of the *worst-case* arrows to lower standard deviations, uni-modal models (left) versus max-mixture models (right), highlights the improvements in modeling capabilities, specifically robustness to outliers.

### C. Learned Weights

We next present a few learned parameter settings for discussion purposes correspond to the respective models shown in Table I.

The constant models learned weights of [9.29] for uni-modal and [[5.2], [16.2]] with  $\alpha = [0.98, 0.02]$  for the max-mixture model. This reflects the fact that the sensor performs well most of the time, but a uni-modal model must compensate for the high-error observations with an overestimate of  $\sigma$  to fit the Gaussian assumption.

We were pleasantly surprised by the quality of the vendor’s uncertainties both in terms of likelihood and in terms of  $\max e_i/\sigma_i$ . The vendor model learned weights of [0.8] and [[0.73], [1.16]] with  $\alpha = [0.97, 0.03]$ , reflecting only modest adjustment of their estimates. However, we were still able to improve upon them with our method.

## VII. CONCLUSION

We have described a general approach for computing sensor uncertainty estimates using a machine learning approach. Feature vectors are constructed from observations and a

weight vector is learned from a ground-truthed data set, via maximizing the log-likelihood of the training data set.

We showed how this approach can be extended to more expressive error models using max-mixtures. We take the view in this paper that “outliers” arise from mismatches between the empirical performance of a system and its error model: better models assign higher probabilities to outliers and thus mitigate their impact. Versus an explicit outlier rejection phase, the max-mixture approach both provides an integrated Bayesian mechanism for robust estimation and removes the need for a near-perfect outlier detector.

In evaluating the performance of these models we use standard log-likelihood metrics and introduce a metric that reflects the impact of an outlier on a SLAM system. Our work was evaluated on a large multi-robot dataset and we demonstrated significant performance improvements using our methods on the task of characterizing error-prone customer-grade GPS sensors. Related methods attempting to add robustness to the back-end of SLAM would also benefit from improved robustness on the front-end provided by the approach presented here.

#### ACKNOWLEDGEMENTS

This work was supported by U.S. DoD Grant FA2386-11-1-4024.

#### REFERENCES

- [1] E. Olson, “Robust and efficient robotic mapping,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.
- [2] N. Sunderhauf and P. Protzel, “Switchable constraints for robust pose graph SLAM,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1879–1884.
- [3] Y. Latif, C. Cadena, and J. Neira, “Realizing, reversing, recovering: Incremental robust loop closing over time using the iRRR algorithm,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4211–4217.
- [4] E. Olson and P. Agarwal, “Inference on networks of mixtures for robust robot mapping,” in *Proceedings of Robotics: Science and Systems (RSS)*, July 2012.
- [5] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, “Robust map optimization using dynamic covariance scaling,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [6] Y. Latif, C. Cadena, and J. Neira, “Robust loop closing over time,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2012.
- [7] P. J. Huber, “Robust estimation of a location parameter,” *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.
- [8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [9] S. Thrun and M. Montemerlo, “The GraphSLAM algorithm with applications to large-scale mapping of urban structures,” *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–430, May-June 2006.
- [10] A. Stentz, C. Dima, C. Wellington, H. Herman, and D. Stager, “A system for semi-autonomous tractor operations,” *Autonomous Robots*, vol. 13, no. 1, pp. 87–104, 2002.
- [11] J. Kim and S. Sukkarieh, “SLAM aided GPS/INS navigation in GPS denied and unknown environments,” in *Proceedings of the 2004 International Symposium on GNSS/GPS*, 2004, pp. 1–5.
- [12] M. Agrawal and K. Konolige, “Real-time localization in outdoor environments using stereo vision and inexpensive GPS,” in *Proceedings of the International Conference on Pattern Recognition*. IEEE, 2006.
- [13] W. Ding, J. Wang, S. Han, A. Almagbile, M. Garratt, A. Lambert, and J. Wang, “Adding optical flow into the GPS/INS integration for uav navigation,” in *Proc. of International Global Navigation Satellite Systems Society Symposium*. Citeseer, 2009, pp. 1–13.
- [14] K. Ohno, T. Tsubouchi, B. Shigematsu, and S. Yuta, “Differential GPS and odometry-based outdoor navigation of a mobile robot,” *Advanced Robotics*, vol. 18, no. 6, pp. 611–635, 2004.
- [15] D. C. Moore, A. S. Huang, M. Walter, E. Olson, L. Fletcher, J. Leonard, and S. Teller, “Simultaneous local and global state estimation for robotic navigation,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 3794–3799.
- [16] L. Cremean, T. Foote, J. Gillula, G. Hines, D. Kogan, K. Kriechbaum, J. Lamb, J. Leibs, L. Lindzey, C. Rasmussen, *et al.*, “Alice: An information-rich autonomous vehicle for high-speed desert navigation,” *Journal of Field Robotics*, vol. 23, no. 9, pp. 777–810, 2006.
- [17] J. Wang, M. Garratt, A. Lambert, J. Wang, S. Han, and D. Sinclair, “Integration of GPS/INS/vision sensors to navigate unmanned aerial vehicles,” in *XXI Congress of the Int. Society of Photogrammetry & Remote Sensing, Beijing, PR China*, 2008, pp. 963–970.
- [18] B. Ben-Moshe, E. Elkin, H. Levi, and A. Weissman, “Improving accuracy of GNSS devices in urban canyons,” in *CCCG*, 2011.
- [19] D. Maier and A. Kleiner, “Improved GPS sensor model for mobile robots in urban terrain,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4385–4390, 2010.
- [20] E. Takeuchi, M. Yamazaki, K. Ohno, and S. Tadokoro, “GPS measurement model with satellite visibility using 3d map for particle filter,” in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. IEEE, 2011, pp. 590–595.
- [21] N. Sunderhauf, M. Obst, G. Wanielik, and P. Protzel, “Multipath mitigation in GNSS-based localization using robust optimization,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 784–789.
- [22] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goedel, and M. Bulic, “Progress towards multi-robot reconnaissance and the MAGIC 2010 Competition,” *Journal of Field Robotics*, vol. 29, September 2012.